Development API Guide

开发 API 指南

I-White-label integration
外包合作对接文档

Version	Author	Descriptions	Approval	Date
Draft		First Draft		03/14/2017
Draft 0.0.3		Changes: - dataTo -> dateTo - change deposit/withdraw: ▪ method PUT -> POST ▪ payload -> parameters - Add more error message: 307, 308, 309, 405, 406 - Remove error message: 303, 304		04/11/2017
Draft 0.0.4		Changes: - Add basic workflow - Add Period status table - Update parameter, data for wager(add parlay, virtual sport)		04/14/2017
Draft 0.0.5		Changes: - Rename balance to availableBalance - Add more outstanding - Remove some redundant fields in wagers - Add more “filterBy” input param in wager service		04/19/2017
Draft 0.0.6 release 5.3		Changes: - Remove VS Product in - Add more data for Outright: parentEventName, category - 2.4 Generate Token: add sample code and explain timestamp - Add header for every services to clearly		05/04/2017
Draft 0.0.7		Changes: - Add more one service wager - Add sample code by C#		05/11/2017
Draft 0.0.8		Changes: - Support more loginId in request/response		05/15/2017
Draft 0.0.9		Changes: - Add more a service “Change Status Member”		05/17/2017

Draft 0.0.10		Changes: - Add Login service v2		05/30/2017
Draft 0.0.11		Changes: - Add WinLoss Simple		06/05/2017
Draft 0.0.12		Changes: - Add Extension for Login and LoginV2		06/05/2017
Draft 0.0.13		Changes: - Add locale param on Wager and All Wager - Add returned settled date (settleDateFm) on wager information - Support new date time format on API Wager and All Wager - Add API Get My Bet		08/23/2017
Draft 0.0.14		Changes: - Add Rule validate loginId when creating a new player for API LoginV2 and Create User - Add Wager Feed		10/12/2017
Draft 0.0.15		- Add Announcement - Add 'turnover' on Wager information		12/05/2017
Draft 0.0.16		Changes: - Add data for teaser wager of FR001, FR002 and FR005 - Add "Teaser" betType in BetType table		05/03/2018
Draft 0.0.17		Changes: - Add sample code for PHP version 7.1 later - Support new param oddsFormat for login and loginV2		06/05/2018
Draft 0.0.18		Changes: - Support new param wagerIds for FR002 - Add Transactions FR007		08/15/2018
Draft 0.0.19		Changes: - Correct translation for ML_1x2 and OU at Bet type (part 5.3)		10/18/2018

		- Add table Selection Type (part 5.4)		
Draft 0.0.20		Changes: - Support new param transactionId for deposit/withdraw if use support recheck. - Add FR008 – Check Status For Deposit/Withdraw - Update more info for Period Status appendix		11/20/2018
Draft 0.0.21		Changes: - Add error code 116 - Selection Type: Adding translation in Chinese - Period status: Adding translation in Chinese - Language: Adding 2 new languages		1/23/2019
Draft 0.0.22		Changes: - Support new param sport for FR006		03/04/2019
Draft 0.0.23		Changes: - Support new validation update_date for param filterBy for FR002		06/05/2019
Draft 0.0.24		Changes: - Add sample code for C# language - Add new API to get hot events		07/19/2019
Draft 0.0.25		Changes: - Hot events API support filter by sports		07/23/2019
Draft 0.0.26		Changes: - Increase loginId's suffix max length to 16, apply for FA003 and FP001		08/01/2019
Draft 0.0.27		Changes: - Add new sports (from '*alpine-skiing*' to '*water-polo*') to Sport list		08/05/2019

1 TABLE OF CONTENTS 目录

1 PURPOSE 目的

I-White-label 3rd Party integration API to allow B2B customers to integrate sportsbook website to B2B customer website. BSB customer's users will be able to access and place bet in sportsbook system from B2B customer's websites without forcing user to register in sportsbook website.

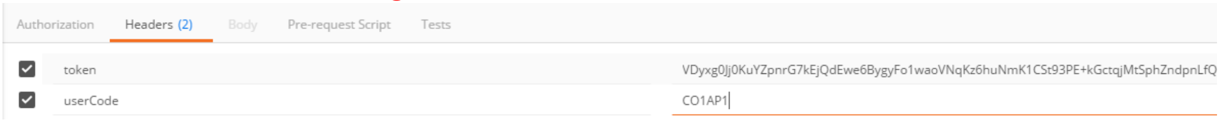
I-外包合作第三方对接 API 能使 B2B 客户将体育对接到 B2B 客户的网站，B2B 客户的用户在不需要在体育平台网站注册用户的情况上能够从 B2B 客户的网站拜访并在体育博彩系统进行投注。

- Create player account 创建用户账户
- Login player 用户登入
- Logout player 用户登出
- Adjust player balance 调整用户余额
- Get Player wagers 获取用户投注

2 BACKGROUND 背景

2.1 AUTHENTICATION & API REQUIREMENTS 认证和 API 要求

1. To be able to call API, B2B customer need to provide IPs to be whitelisted.
   为了能够呼唤 API, B2B 客户需要提供 IP 供平博技术团队列入白名单。
2. All functional call API are passed authentication in HTTP Header:
   所有的功能调用 API 都通过 HTTP Header 中的认证:
   ✓ token: <token>
   ✓ userCode: <agent code>



✓	token	VDyxg0Jj0KuYZpnrG7kEjQdEwe6BygyFo1waoVNqKz6huNmK1CSt93PE+kGctqjMtSphZndpnLfQ
✓	userCode	CO1AP1

3. You need have 3 important items as below to generate:
   您需要产生 3 个重要的项目如下
   ✓ Agent code (**agentCode**)
   ✓ Agent key (**agentKey**)
   ✓ Secret key (**secretKey**)

2.2 DATA FORMATS 数据格式

API supports only JSON format.
API 只支持 JSON 格式

2.3 DATE TIME 日期时间

All date will be in GMT-4 except Timestamp

除时间戳 Timestamp 以外， 所有日期均为 GMT-4Date format: yyyy-MM-dd

日期格式：年-月-日

Date Time format: yyyy-MM-dd HH:mm:ss. For example: 2017-04-01 07:00:00

日期时间格式：年-月-日 小时：分钟：秒钟。 例如： 2017-04-01 07:00:00

Timestamp (only use for generate token): A UNIX timestamp, also known as Epoch Time or POSIX timestamp, is a representation of a moment defined as the time that has elapsed since a reference known as the UNIX epoch: 1970-01-01 00:00:00 UTC (what is UTC). Using Java as an example, System.currentTimeMillis() returns just that, a UNIX timestamp in milliseconds.

时间戳（仅用于产生代码）： UNIX 时间戳，也称为"Epoch Time"或 POSIX 时间戳， 定义为 UNIX epoch 是一个时间的表示已经过去的时间：1970-01-01 00:00:00 UTC（什么是 UTC）。以 Java 为例，System.currentTimeMillis()返回一个 UNIX 时间戳（毫秒）。

Please use this code to get timestamp in java: long timestamp = System.currentTimeMillis();

请用这个代码在 java 获取时间戳: long timestamp = System.currentTimeMillis();

For example: 1493366020081

例如：1493366020081

2.4 GENERATE TOKEN 产生代码

Timestamp in milliseconds format (*A UNIX timestamp, also known as Epoch Time or POSIX timestamp*);

时间戳，以毫秒为单位(UNIX 时间戳, 也称为 Epoch Time 或 POSIX 时间戳)

String hashToken = md5(agentCode + timestamp + agentKey);
String tokenPayLoad = agentCode|timestamp|hashToken;
String token = encryptAES(secretKey, tokenPayLoad);

Sample code (java) 示例代码 (java)

```java
import java.security.NoSuchAlgorithmException;
import java.util.Base64;
import java.util.concurrent.TimeUnit;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.crypto.Cipher;
import javax.crypto.spec.IvParameterSpec;
import javax.crypto.spec.SecretKeySpec;
import org.apache.commons.codec.digest.DigestUtils;

private static final String ALGORITHM = "AES";
private static final String INIT_VECTOR = "RandomInitVector";
private static String generateToken(String agentCode, String agentKey, String secretKey) throws
 NoSuchAlgorithmException {
    String sTimestamp = String.valueOf(System.currentTimeMillis());
    String hashToken = DigestUtils.md5Hex(agentCode + sTimestamp + agentKey);
    String tokenPayLoad = String.format("%s|%s|%s", agentCode, sTimestamp, hashToken);
    String token = encryptAES(secretKey, tokenPayLoad);
    return token;
}

private static String encryptAES(String secretKey, String tokenPayLoad) {
    try {
        SecretKeySpec secretKeySpec = new SecretKeySpec(secretKey.getBytes(), ALGORITHM);
        IvParameterSpec iv = new IvParameterSpec(INIT_VECTOR.getBytes("UTF-8"));
        Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5PADDING");
        cipher.init(Cipher.ENCRYPT_MODE, secretKeySpec, iv);
        byte[] encrypted = cipher.doFinal(tokenPayLoad.getBytes());
        return Base64.getEncoder().encodeToString(encrypted);
    } catch (Exception ex) {
        logger.log(Level.INFO, "An exceptionn occurred when encript key: {0}, plain text: {1}, exception {2}", new Object[]{secretKey, tokenPayLoad, ex.toString()});
    }
    return null;
}
```

Sample code (C#)示例代码(C#)

```csharp
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Security.Cryptography;

private static string INIT_VECTOR = "RandomInitVector";

public string generateToken(string agentCode, string agentKey, string secretKey)
{
    string timeStamp = ((long)(DateTime.UtcNow - new DateTime(1970, 1, 1)).TotalMilliseconds).ToString();
    string hashToken = md5Hex(agentCode + timeStamp + agentKey);
    string tokenPayLoad = String.Format("{0}|{1}|{2}", agentCode, timeStamp, hashToken);
    return encryptAES(secretKey, tokenPayLoad);
}
private string md5Hex(String data)
{
    MD5CryptoServiceProvider x = new MD5CryptoServiceProvider();
    byte[] bs = System.Text.Encoding.UTF8.GetBytes(data);
    bs = x.ComputeHash(bs);
    StringBuilder s = new StringBuilder();
    foreach (byte b in bs)
    {
        s.Append(b.ToString("x2").ToLower());
    }


    return s.ToString();
}
public string encryptAES(string secretKey, string tokenPayLoad)
{
    using (
        var csp = new AesCryptoServiceProvider())
    {
        ICryptoTransform e = GetCryptoTransform(csp, true, secretKey);
        byte[] inputBuffer = Encoding.UTF8.GetBytes(tokenPayLoad);
        byte[] output = e.TransformFinalBlock(inputBuffer, 0, inputBuffer.Length);
        string encrypted = Convert.ToBase64String(output);
        return encrypted;
    }
}
private ICryptoTransform GetCryptoTransform(AesCryptoServiceProvider csp, bool encrypting, string secretKey)
{
    csp.Mode = CipherMode.CBC;
    csp.Padding = PaddingMode.PKCS7;
    byte[] key = Encoding.UTF8.GetBytes(secretKey);

    csp.IV = Encoding.UTF8.GetBytes(INIT_VECTOR);
    csp.Key = key;
    if (encrypting)
    {
        return csp.CreateEncryptor();
    }
    return csp.CreateDecryptor();
}
```

Sample Code (PHP version 7.0 before) 示例代码（PHP 7.0 版本以前）

This sample code is suitable for PHP version from 7.0.x before.

```php
<?php
class PHPCipher {

        public function generateToken($agentCode, $agentKey, $secretKey)
        {
            $timestamp = time()*1000;
            $hashToken = md5($agentCode. $timestamp . $agentKey);
            $tokenPayLoad = $agentCode . '|' . $timestamp . '|' . $hashToken;
            $token = $this->encryptAES($secretKey, $tokenPayLoad);

            return $token;
        }
        private function encryptAES($secretKey, $tokenPayLoad)
        {
            $size = mcrypt_get_block_size(MCRYPT_RIJNDAEL_128, MCRYPT_MODE_CBC);
            $td = mcrypt_module_open(MCRYPT_RIJNDAEL_128, '', MCRYPT_MODE_CBC, '');

            $iv = "RandomInitVector";
            mcrypt_generic_init($td, $secretKey, $iv);
            $encrypt = mcrypt_generic($td, $this->pad($tokenPayLoad, $size));
            mcrypt_generic_deinit($td);
            mcrypt_module_close($td);
            return base64_encode($encrypt);
        }

        private function pad($text, $blocksize)
        {
            $pad = $blocksize - (strlen($text) % $blocksize);
            return $text . str_repeat(chr($pad), $pad);
        }
    }


    $agentCode = "CO1AP1";
    $agentKey = "RR3AaKBxGxLqEFpCXh0O3nJBsqRgDWA9nFU1SYEeLgmt123456";
    $secretKey = "1234567890123456";


    $cipher = new PHPCipher();

    echo $cipher->generateToken($agentCode, $agentKey, $secretKey);


?>
```

Sample Code (PHP version 5.x, 6.x and 7.1 later) 示例代码（PHP 版本 5.x, 6.x and 7.1 及更新版本）

This sample code is suitable for almost PHP version from version 5.x to version 7.1.x later. You need active extension `php_openssl`

以下示例代码适用于 PHP5.x 到 7.1.x 版本以及更新版本。您需要用到活跃后缀 `php_openssl`

```php
<?php

class PHPCipher
{

    public function generateToken($agentCode, $agentKey, $secretKey)
    {
        $timestamp = time()*1000;
        $hashToken = md5($agentCode. $timestamp . $agentKey);
        $tokenPayLoad = $agentCode . '|' . $timestamp . '|' . $hashToken;
        $token = $this->encryptAES($secretKey, $tokenPayLoad);

        return $token;
    }

    private function encryptAES($secretKey, $tokenPayLoad)
    {
        $iv = "RandomInitVector";
        $encrypt = openssl_encrypt($tokenPayLoad, "AES-128-CBC", $secretKey, OPENSSL_RAW_DATA, $iv);

        return base64_encode($encrypt);
    }

}

$agentCode = "C01AP1";
$agentKey = "RR3AaKBxGxLqEFpCXh0O3nJBsqRgDWA9nFU1SYEeLgmt123456";
$secretKey = "1234567890123456";

$cipher = new PHPCipher();

echo $cipher->generateToken($agentCode, $agentKey, $secretKey);
?>
```

2.5 Errors and Exception 错误和例外

Error response with known error.

已知错误的错误响应

```
{
      code: '108',
      message: 'Agent invalid. Please contact partner to get agent code.'
}
```

Please see details at *Error Table*

请在错误表格中参看详细信息

Unexpected exception error format only including “trace”. Customer can send trace_code to IT Support Team for helping.

意外的异常错误格式只会包含 “追踪 trace”代码. 客户可以发送追踪代码 trace_code 给技术服务团队申请协助。

```
{
      trace: 'C01AP1|VDyxg0Jj0KuYZpnrG7kEjYaVZDfieeDTa4Fhb+v1ZHjWoKY4a2t2PJNOX/A88+vLLl2fD0ep1hK6cwmj0z0fEA==|28642330553954',
}
```

2.6 Getting Started 如何开始

Step 1 – Sign up

步骤 1 – 注册
To get started you need to create an account and get agent code via agent site.
开始前，您需要创建一个帐户，并通过代理网站获取 agent code。

Step 2 – Receive agent key
步骤 2 – 获得 agentKey
You need send **agentCode** (create at step 1) to our Partner Support Center to get **agentKey** and **secretKey**
您需要将 **agentCode**（步骤 1 创建）发给我们代理服务中心来获取 **agentKey** 和 **secretKey**
Please note that your request to access API the IP must in our white list.
请注意：IP
需要先列入白名单，您才能够申请访问

Step 3 – Generate token (refer 2.4)
步骤 3 – 产生代码（参考 2.4）

Step 4 – Call API service
步骤 4 – 呼唤 API 服务
For each service call, you need pass authentication token and suitable parameters
对每个服务呼唤，您需要通过代码验证以及的合适的参数

3 API FUNCTIONS API 功能

3.1 FA001 – Login 登入

This service is used to generate URL by pass login for player
这个服务是让用户通过 pass login 来产生 URL

Endpoint 端点

Name	Value	Description
URL	/player/login	
Method	GET/POST	

Headers 头文件

Name	Value	Validation	Description
userCode	String (required 必需项)		This is agent code that you get at step 2. E.g: CO1AP1

			此为在第二步骤获取的代理编号，例如，CO1AP1
token	String (required 必需项)	*Token is available in **15 minutes** since it was created* *令牌在创建之后的 15 分钟内有效*	

Parameters 参数

Name	Value	Validation	Description
userCode	String (required 必需项)		This is user code / loginID of player. E.g: PA10000000 此为玩家登录名/用户名，例如 PA10000000
locale	String (optional 非必需项)	List supported locales based on available brand's languages.	Appendix Locale (Language)
oddsFormat	String (optional 非必需项)	List supported oddsFormat: AM, EU, HK, ID, MY	Appendix Odds Format

Workflow 工作流程

To be added 以后添加

Format URL login URL 登录格式

<host>/member/<locale>/sports?oddsFormat=HK&token=<token>

Example:

http://wlqat.oreo88.com/member/en/sports?oddsFormat=HK&token=TkIxS3pJc2RwdituRmNXam1NVGk0ZlhsZ0lrV285Y2JncFAvVXN2V2dsNVAvNHdXejFadHQ1UTZoeFB6c201cXQ4dXc5NC8vL0pvNHhDQUN4bng0T2c9PQ==

***Note:** this token generated by the Pinnacle system for player to login to the system. It is not the token that 3rd party generate.*

***注意**: 这个代码是由平博系统产生让用户登入我们的系统。 这个不是第三方产生的代码。*

Sample code (java) - *See HttpUtils class at Appendix*

示例代码（java） – 请参阅附录里的 HttpUtils class

```
import java.io.IOException;
import java.util.HashMap;
import java.util.Map;
```

```java
private static void getTokenLogin() throws IOException {
    Map params = new HashMap();
    params.put("userCode", "PA10000000");
    params.put("locale", "zh-cn");
    params.put("oddsFormat", "HK");
    Map headers = new HashMap();
    headers.put("userCode", "CO1AP1");
    headers.put("token", "gvbLZb70DhMNXMLw3egvktPT3sfDWnDC5QI97MXHqK9FCFO1n6oepaXQivCIVpsDlth/jbR/qgUcpi2wHc62Tw==");

    String url = "http://paapiqat.oreo88.com/b2b/player/login";
    String result = HttpUtils.post(url, "", params, headers);
    System.out.println(result);
}
```

Sample code (C#) 示例代码（C#）

```csharp
using System;
using System.Net.Http;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace Login
{
    class Program
    {
        static void Main(string[] args)
        {
            Dictionary<string, string> parameters = new Dictionary<string, string>();
            parameters.Add("userCode", "QAT0101.05");
            parameters.Add("locale", "zh-cn");
            parameters.Add("oddsFormat", "HK");

            Dictionary<string, string> headers = new Dictionary<string, string>();
            headers.Add("userCode","QAT0101");
            headers.Add("token","LrSnxY89VuOx05LD6goEQ5fcnWzKT36mwhdJ73A86WJO66wzGmJN9YGDgR8Ac9qLWCTmpaOtZUgv2r714TiZGQ==");

            Console.WriteLine(HttpUtils.PostRequest("https://api.ps3838.com/b2b/player/login", parameters, headers).Result);
            Console.ReadKey();
        }
    }
    class HttpUtils
    {
        public async static Task<string> PostRequest(String url, Dictionary<string, string> parameters, Dictionary<string, string> headers)
        {
            HttpClient client = new HttpClient();
            foreach(var item in headers) {
                client.DefaultRequestHeaders.Add(item.Key, item.Value);
            }

            var content = new FormUrlEncodedContent(parameters);
            var response = await client.PostAsync(url + "?" + content.ReadAsStringAsync().Result, null);
            return await response.Content.R     eadAsStringAsync();
        }
        public async static Task<string> GetRequest(String url, Dictionary<string, string> parameters, Dictionary<string, string> headers)
        {
            HttpClient client = new HttpClient();
            foreach(var item in headers){ client.DefaultRequestHeaders.Add(item.Key, item.Value);}

            var content = new FormUrlEncodedContent(parameters);
            var response = await client.GetAsync(url + "?" + content.ReadAsStringAsync().Result);
            return await response.Content.ReadAsStringAsync();
        }
    }
}
```

Response 反应

Result is URL to login System. Then we use this url to open new popup in browser. This url will auto login the system.

结果是登录系统的 URL。然后我们使用这个 URL 在浏览器中打开新的弹出窗口。这个网址会自动登录系统。

```
{
  loginUrl: 'http://authen.pinbet88.com/member/en?oddsFormat=HK&token=xyz',
  userCode: 'PA10000000',
  loginId: 'PA10.0',
  token: 'thisislogintokengeneratetologin',
  updatedDate: 1459312108353
}
```

Extension 扩充

In Mobile version we have support “Back” icon to redirect to [custom URL]. Here [custom URL] is the url that you add more “externalUrl” param to Login link responded.

在移动版，我们有“返回”的图标重定向到[定制 URL]. 这里的[定制 URL]是你多加的“externalURL”参数给登陆链接回应。

For example:
http://wlqat.oreo88.com/en/sports?token=eGRDMFRQZDZQYlhIelBFdXZ2UVN5Nm9MNHlkNThVc0JiRm5SN2hMcVE3dEhHZm00ZnFDbmVMYzB5OTVFYW1NV0NZN0F2c0tqZXFjSU9EdjJhN0tiWXc9PQ==&externalUrl=http://google.com

When you click on “Back” icon, app will send redirect to google.com page

当你点击“返回”图标，应用程序将发送重定向到 google.com 页面

***Note:** this feature is only available on Mobile version*

注意：此功能只有在移动版本上有

3.2 FA002– Logout 登出

If you need to logout from system by userCode. You need call this function.

如果你需要通过 userCode 登出系统。你需要调用这个函数。

Endpoint 端点

Name	Value	Description
URL	/player/logout	
Method	GET/POST	

Headers 头文件

Name	Value	Validation	Description
userCode	String (required 必需项)		This is agent code that you get at step 2. E.g: CO1AP1 此为在第二步骤获取的代理编号，例如，CO1AP1
token	String (optional 非必需项)	*Token is available in **15 minutes** since it was created 令牌在创建之后的 15 分钟内有效*	

Parameters 参数

Name	Value	Validation	Description
userCode	String (required 必需项)		This is user code / loginID of player . E.g: PA10000000 此为玩家登录名/用户名，例如 PA10000000

Sample code (java) - *See HttpUtils class at Appendix*
示例代码（java） – 请参阅附录里的 HttpUtils class

```java
import java.io.IOException;
import java.util.HashMap;
import java.util.Map;

private static void doLogout() throws IOException {
    Map params = new HashMap();
    params.put("userCode", "PA10000000");
    Map headers = new HashMap();
    headers.put("userCode", "CO1AP1");
    headers.put("token", "gvbLZb70DhMNXMLw3egvktPT3sfDWnDC5QI97MXHqK9FCFO1n6oepaXQivCIVpsDlth/jbR/qgUcpi2wHc62Tw==");

    String url = "http://paapiqat.oreo88.com/b2b/player/logout";
    String result = HttpUtils.post(url, "", params, headers);
    System.out.println(result);
}
```

Sample code (C#) 示例代码（C#）

```
using System;
using System.Net.Http;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace Logout
{
    class Program
    {
        static void Main(string[] args)
        {
            Dictionary<string, string> parameters = new Dictionary<string, string>()
;
            parameters.Add("userCode", "PA10000000");

            Dictionary<string, string> headers = new Dictionary<string, string>();
            headers.Add("userCode","CO1AP1");
            headers.Add("token","DIwYQJZIYiosWrbXQe+TFdyMk6POZvbcM1KjxxQObZEn0+efzoM
Cb3i+PWr1ZFuj0UciR8w+qqo1M2hJ965Y9w==");

            Console.WriteLine(HttpUtils.PostRequest("http://paapiqat.oreo88.com/b2b/
player/logout", parameters, headers).Result);
            Console.ReadKey();
        }
    }
}
```

Response 反应

```
{
"status": "successful"
}
```

3.3 FA003 – LoginV2 登陆 V2

This service is used to create new player user and generate URL by pass login for player. This service is difference FA001 is in case if player hasn't existed in system, new user will be created.

这项服务是用以创建新用户并对透过 pass login 的用户产生 URL。 这个服务跟 FA001 不同的是如果用户在系统里不存在，将会创建新用户

Endpoint 端点

Name	Value	Description
URL	/player/loginV2	
Method	GET/POST	

Headers 头文件

Name	Value	Validation	Description
userCode	String (required 必需项)		This is agent code that you get at step 2. E.g: CO1AP1

			此为在第二步骤获取的代理编号，例如，CO1AP1
token	String (required 必需项)	*Token is available in **15 minutes** since it was created 令牌在创建之后的 15 分钟内有效*	

Parameters 参数

Name	Value	Validation	Description
loginId	String (required 必需项)	If loginId has not existed in system, it will validate and create new player based on this loginId. **Rule validate loginId:** loginId format is ***prefix.suffix*** Must be between 6 and 16 alphanumeric characters and number in ***Suffix***. Must have one dot (.) in between ***Prefix*** and ***Suffix***. Total length of loginId is not more than 50 alphanumeric characters.	This is user code / loginID of player. E.g: PA10000000 or PA10.abc123 此为玩家登录名/用户名，例如 PA10000000 或 PA10.abc123
locale	String (optional 非必需项)	List supported locales based on available brand's languages.	Appendix Locale (Language)
sport	String (optional 非必需项)	List Sport supported	Appendix Sport
oddsFormat	String (optional 非必需项)	List supported oddsFormat: AM, EU, HK, ID, MY	Appendix Odds Format

Workflow工作流程

To be added
以后添加

Format URL login URL登陆格式

<host>/member-service/v1/login-token?locale=<en>&sport=<sport>&oddsFormat=<oddsFormat>&token=<token>
Example:

http://wlqat.oreo88.com/member-service/v1/login-token?locale=en&sport=soccer&oddsFormat=HK&token=YkZXSFl5S0VsUElrQzRlRS9ZRERsbDN3dktnb2dnSE1tcTZ1VTEvVnFldTJkVTcyWklJeVVRSk1xY3dpM2VWVXFYY3IxMzRpWGNBWllsakV2Wk1JZ0E9PQ==

Note: this token generated by our system for player login to our system. It is not token that 3rd party generate.

注意：这个代码是由我们的系统生成，供用户登录我们的系统。这不是由第三方生成的。

Sample code (java) - *See HttpUtils class at Appendix*

示例代码（java）– 请参阅附录里的 HttpUtils class

```java
import java.io.IOException;
import java.util.HashMap;
import java.util.Map;
private static void getTokenLogin() throws IOException {
    Map params = new HashMap();
    params.put("loginId", "PA10.02");
    params.put("locale", "zh-cn");
    params.put("sport", "tennis");
    Map headers = new HashMap();
    headers.put("userCode", "CO1AP1");
    headers.put("token", "gvbLZb70DhMNXMLw3egvktPT3sfDWnDC5QI97MXHqK9FCFO1n6oepaXQivCIVpsDlth/jbR/qgUcpi2wHc62Tw==");

    String url = "http://paapiqat.oreo88.com/b2b/player/loginV2";
    String result = HttpUtils.post(url, "", params, headers);
    System.out.println(result);
}
```

Sample code (C#) 示例代码（C#）

```csharp
using System;
using System.Net.Http;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace LoginV2
{
    class Program
    {
        static void Main(string[] args)
        {
            Dictionary<string, string> parameters = new Dictionary<string, string>();
            parameters.Add("loginId", "PA10.02");
            parameters.Add("locale", "zh-cn");
            parameters.Add("sport", "tennis");

            Dictionary<string, string> headers = new Dictionary<string, string>();
            headers.Add("userCode","CO1AP1");
            headers.Add("token","DIwYQJZIYiosWrbXQe+TFdyMk6POZvbcM1KjxxQObZEn0+efzoMCb3i+PWr1ZFuj0UciR8w+qqo1M2hJ965Y9w==");

            Console.WriteLine(HttpUtils.PostRequest("http://paapiqat.oreo88.com/b2b/player/loginV2", parameters, headers).Result);
            Console.ReadKey();
        }
    }
}
```

Response 反应

Result is URL to login System. Then we use this url to open new popup in browser. This url will auto login Pinbet88 system.
然后我们用这个 URL 在浏览器中打开新的弹出视窗。

```
{
  loginUrl: 'http://wlqat.oreo88.com/member-service/v1/login-token?locale=zh-cn&sport=tennis&token=YkZXSFl5S0VsUElrQzRlRS9ZRERsbDN3dktnb2dnSE1tcTZ1VTEvVnFldTJkVTcyWklJeVVRSk1xY3dpM2VWVXFYY3IxMzRpWGNBWllsakV2Wk1JZ0E9PQ==,
  userCode: 'PA10000001',
  loginId: 'PA10.02',
  token: 'YkZXSFl5S0VsUElrQzRlRS9ZRERsbDN3dktnb2dnSE1tcTZ1VTEvVnFldTJkVTcyWklJeVVRSk1xY3dpM2VWVXFYY3IxMzRpWGNBWllsakV2Wk1JZ0E9PQ==',
  updatedDate: '2017-05-26 05:37:16'

}
```

Extension 扩充

In Mobile version we have support "Back" icon to redirect to [custom URL]. Here [custom URL] is the url that you add more "externalUrl" param to Login link responded.

在移动版，我们有"返回"的图标 重定向[定制 URL]. 这里的[定制 URL]是你多加的"externalURL"参数给登陆链接回应。

For example:

http://wlqat.oreo88.com/member-service/v1/login-token?locale=en&token=eGRDMFRQZDZQYlhIelBFdXZ2UVN5NmZNWXE4RFlnUm1vS1pFeVFnVFV4MEpWYkZ5SlczM0ZmZTlFNUhlTytRYTdXeUNmUTN2ak5iVXpQbTNLVWpCUkE9PQ==&externalUrl=http://google.com

When you click on "Back" icon, app will send redirect to google.com page
当你点击"返回"图标，应用程序将发送重定向到 google.com 页面

Note: this feature has only on Mobile version
注意：此功能只有在移动版本上有

3.4 FP001 – Create User 创建用户

This service is used to sign up a player
这个服务是用来注册用户

Endpoint 端点

Name	Value	Description

URL	/player/create	
Method	GET/POST	

Headers 头文件

Name	Value	Validation	Description
userCode	String (required 必需项)		This is agent code that you get at step 2. E.g: CO1AP1 此为在第二步骤获取的代理编号，例如，CO1AP1
token	String (required 必需项)	*Token is available in **15 minutes** since it was created 令牌在创建之后的 15 分钟内有效*	

Parameters 参数

Name	Value	Validation	Description
agentCode	String (optional 非必需项)		Agent code downline that new created-player will be belong to. If no agent code provided, player will be created direct under agent code calling this API. E.g: CO1AP100 新建玩家账户所属的代理编号。如果没有提供代理编号，那么玩家用户会直接创建在调用此 API 的代理账户之下，例如：CO1AP100
loginId	String (optional 非必需项)	If you input loginId, system will validate loginId and response error if it exist in system.	**Rule validate loginId:** loginId format is ***prefix.suffix*** Must be between 6 and 16 alphanumeric characters and number in ***Suffix***. Must have one dot (.) in between ***Prefix*** and ***Suffix***.

			Total length of loginId is not more than 50 alphanumeric characters. 登录名验证条件: 必须有前缀和后缀 后缀必须在 6-16 个字母和数字之间 必须有一点将前缀和后缀分开 登录名不得长于 50 个字母

Sample code (java) - *See HttpUtils class at Appendix*

示例代码（HttpU– 请参阅附录里的 HttpUtils class

```java
private static void createNewUser() throws IOException {
    Map params = new HashMap();
//      params.put("agentCode", "CO1AP100");   //optional parameter
//      params.put("loginId", "PA10000001");   //optional parameter
    Map headers = new HashMap();
    headers.put("userCode", "CO1AP1");
    headers.put("token", "gvbLZb70DhMNXMLw3egvktPT3sfDWnDC5QI97MXHqK9FCFO1n6oepaXQivCIVpsDlth/jbR/qgUcpi2wHc62Tw==");

    String url = "http://paapiqat.oreo88.com/b2b/player/create";
    String result = HttpUtils.post(url, "", params, headers);
    System.out.println(result);
}
```

Sample code (C#) 示例代码（C#）

```csharp
using System;
using System.Net.Http;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace CreateUser
{
    class Program
    {
        static void Main(string[] args)
        {
            Dictionary<string, string> parameters = new Dictionary<string, string>();

            Dictionary<string, string> headers = new Dictionary<string, string>();
            headers.Add("userCode","CO1AP1");
            headers.Add("token","DIwYQJZIYiosWrbXQe+TFdyMk6POZvbcM1KjxxQObZEn0+efzoMCb3i+PWr1ZFuj0UciR8w+qqo1M2hJ965Y9w==");

            Console.WriteLine(HttpUtils.PostRequest("http://paapiqat.oreo88.com/b2b/player/create", parameters, headers).Result);
            Console.ReadKey();
        }
    }
}
```

Response 反应

```
{
  userCode: 'PA10000000',    // String [successful, fail]
  loginId: 'PA10.0',
}
```

3.5 FP003 – Get Player 获取用户

Get player information.

获取用户资料

Endpoint 端点

Name	Value	Description
URL	/player/info	
Method	GET	

Headers 头文件

Name	Value	Validation	Description
userCode	String (required 必需项)		This is agent code that you get at step 2. E.g: CO1AP1 此为在第二步骤获取的代理编号，例如，CO1AP1
token	String (required 必需项)	*Token is available in **15 minutes** since it was created 令牌在创建之后的 15 分钟内有效*	

Parameters 参数

Name	Value	Validation	Description
userCode	String (required 必需项)		This is user code / loginID of player. E.g: PA10000000 此为玩家登录名/用户名，例如 PA10000000

Sample code (java) - *See HttpUtils class at Appendix*

示例代码（java）– 请参阅附录里的 HttpUtils class

```
import java.io.IOException;
import java.util.HashMap;
import java.util.Map;
```

```java

private static void getPlayer() throws IOException {
Map params = new HashMap();
params.put("userCode", "PA10000000");
Map headers = new HashMap();
headers.put("userCode", "CO1AP1");
headers.put("token", "gvbLZb70DhMNXMLw3egvktPT3sfDWnDC5QI97MXHqK9FCFO1n6oepaXQivCIVpsDlth/jbR/qgUcpi2wHc62Tw==");

String url = "http://paapiqat.oreo88.com/b2b/player/info";
String result = HttpUtils.get(url, "", params, headers);
System.out.println(result);
}
```

Sample code (C#) 示例代码（C#）

```csharp
using System;
using System.Net.Http;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace GetPlayer
{
    class Program
    {
        static void Main(string[] args)
        {
            Dictionary<string, string> parameters = new Dictionary<string, string>();
            parameters.Add("userCode", " PA10000000");

            Dictionary<string, string> headers = new Dictionary<string, string>();
            headers.Add("userCode"," CO1AP1");
            headers.Add("token","UFizq11OTAHxGdaCkzxd1rymMxvIxc8tr3I5oLo3Z5jyh77EpddNrcwHO+omwBnyl1uNrR+xF2+Hyqbw+8eOFA==");

            Console.WriteLine(HttpUtils.GetRequest("http://paapiqat.oreo88.com/b2b/player/info", parameters, headers).Result);
            Console.ReadKey();
        }
    }
}
```

Response 反应

```json
{
  "userCode": "PA10000000",
  "loginId": "PA10.0",
  "firstName": "",
  "lastName": "",
  "status": "ACTIVE", // (User Status)
  "availableBalance": 0,
  "outstanding": 0,
  "createdDate": 2017-04-23 22:24:07,
  "createdBy": "B2B"
}
```

3.6 FP004 – Get List Player 获取用户列表

Get list player information.

获取用户列表信息

Endpoint 端点

Name	Value	Description
URL	/list-player/info	
Method	GET	

Headers 头文件

Name	Value	Validation	Description
userCode	String (required 必需项)		This is agent code that you get at step 2. E.g: CO1AP1 此为在第二步骤获取的代理编号，例如，CO1AP1
token	String (required 必需项)	*Token is available in **15 minutes** since it was created 令牌在创建之后的 15 分钟内有效*	

Parameters *None* **参数** *没有*

Sample code (java) - *See HttpUtils class at Appendix*

示例代码（java） – 请参阅附录里的 HttpUtils class

```
private static void getPlayer() throws IOException {
Map params = new HashMap();
Map headers = new HashMap();
headers.put("userCode", "CO1AP1");
headers.put("token", "gvbLZb70DhMNXMLw3egvktPT3sfDWnDC5QI97MXHqK9FCFO1n6oepaXQivCIVpsDlth/jbR/qgUcpi2wHc62Tw==");

String url = "http://paapiqat.oreo88.com/b2b/list-player/info";
String result = HttpUtils.get(url, "", params, headers);
System.out.println(result);
}
```

Sample code (C#) 示例代码（C#）

```
using System;
using System.Net.Http;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace GetListPlayer
{
    class Program
    {
        static void Main(string[] args)
        {
            Dictionary<string, string> parameters = new Dictionary<string, string>()
;

            Dictionary<string, string> headers = new Dictionary<string, string>();
            headers.Add("userCode","CO1AP1");
            headers.Add("token","DIwYQJZIYiosWrbXQe+TFdyMk6POZvbcM1KjxxQObZEn0+efzoM
Cb3i+PWr1ZFuj0UciR8w+qqo1M2hJ965Y9w==");

            Console.WriteLine(HttpUtils.GetRequest("http://paapiqat.oreo88.com/b2b/l
ist-player/info", parameters, headers).Result);
            Console.ReadKey();
        }
    }
}
```

Response 反应

```
[{
  "userCode": "PA10000000",
  "loginId": "PA10.0",
  "firstName": "",
  "lastName": "",
  "status": "ACTIVE", // (User Status)
  "availableBalance": 0,
  "outstanding": 0,
  "createdDate": 2017-04-23 22:24:07,
  "createdBy": "B2B"
}]
```

3.7 FP005 – DEPOSIT 存款

Deposit for player.

用户存款

Endpoint

Name	Value	Description
URL	/player/deposit	
Method	GET/POST	

Headers 头文件

Name	Value	Validation	Description
userCode	String (required 必需项)		This is agent code that you get at step 2. E.g: CO1AP1 此为在第二步骤获取的代理编号，例如，CO1AP1

token	String (required 必需项)	*Token is available in **15 minutes** since it was created 令牌在创建之后的 15 分钟内有效*	

Parameters 参数

Name	Value	Validation	Description
userCode	String (required 必需项)		This is user code / loginID of player. E.g: PA10000000 此为玩家登录名/用户名，例如 PA10000000
amount	Decimal(10, 2) (required 必需项)	Value > 0	
transactionId	Long (optional 非必需项)	Value > 0	transaction Id you can use to check transaction status and it's unique 用于检查交易状态的交易 ID

Sample code (java) - *See HttpUtils class at Appendix*
示例代码（java） – 请参阅附录里的 HttpUtils class

```java
import java.io.IOException;
import java.util.HashMap;
import java.util.Map;

private static void deposit() throws IOException {
    Map params = new HashMap();
    params.put("userCode", "PA10000000");
    params.put("amount", 10);
    Map headers = new HashMap();
    headers.put("userCode", "CO1AP1");
    headers.put("token", "gvbLZb70DhMNXMLw3egvktPT3sfDWnDC5QI97MXHqK9FCFO1n6oepaXQivCIVpsDlth/jbR/qgUcpi2wHc62Tw==");

    String url = "http://paapiqat.oreo88.com/b2b/player/deposit";
    String result = HttpUtils.post(url, "", params, headers);
    System.out.println(result);
}
```

Sample code (C#) 示例代码（C#）

```csharp
using System;
using System.Net.Http;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace Deposit
{
    class Program
    {
        static void Main(string[] args)
        {
            Dictionary<string, string> parameters = new Dictionary<string, string>()
;
            parameters.Add("userCode", "PA10000000");
            parameters.Add("amount", "10");

            Dictionary<string, string> headers = new Dictionary<string, string>();
            headers.Add("userCode","CO1AP1");
            headers.Add("token","DIwYQJZIYiosWrbXQe+TFdyMk6POZvbcM1KjxxQObZEn0+efzoM
Cb3i+PWr1ZFuj0UciR8w+qqo1M2hJ965Y9w==");

            Console.WriteLine(HttpUtils.PostRequest("http://paapiqat.oreo88.com/b2b/
player/deposit", parameters, headers).Result);
            Console.ReadKey();
        }
    }
}
```

Response 反应

```
{
  'userCode': 'PA10000000',      // string
  'loginId': 'PA10.0',      // string
  'availableBalance': 1010.00,    // decimal(after deposit)
  'amount': 10.00       // decimal
}
```

3.8 FP006 – WITHDRAW 提款

Withdraw for player.

用户提款

Endpoint

Name	Value	Description
URL	/player/withdraw	
Method	GET/POST	

Headers 头文件

Name	Value	Validation	Description
userCode	String (required 必需项)		This is agent code that you get at step 2. E.g: CO1AP1 此为在第二步骤获取的代理编号，例如，CO1AP1
token	String (required 必需项)	*Token is available in **15 minutes** since it was created* *令牌在创建之后的 15 分钟内有效*	

Parameters

Name	Value	Validation	Description
userCode	String (required 必需项)		This is user code / loginID of player. E.g: PA10000000 此为玩家登录名/用户名，例如 PA10000000
amount	Decimal(10, 2) (required 必需项)	Value ＞0	
transactionId	Long (optional 非必需项)	Value > 0	transaction Id you can use to check transaction status and it's unique 用于检查交易状态的交易 ID

Sample code (java) - *[See HttpUtils class at Appendix]*

示例代码（java）－请参阅附录里的 HttpUtils class

```java
import java.io.IOException;
import java.util.HashMap;
import java.util.Map;

private static void withdraw() throws IOException {
    Map params = new HashMap();
    params.put("userCode", "PA10000000");
    params.put("amount", 10);
    Map headers = new HashMap();
    headers.put("userCode", "CO1AP1");
    headers.put("token", "gvbLZb70DhMNXMLw3egvktPT3sfDWnDC5QI97MXHqK9FCFO1n6oepaXQivCIVpsDlth/jbR/qgUcpi2wHc62Tw==");

    String url = "http://paapiqat.oreo88.com/b2b/player/withdraw";
    String result = HttpUtils.post(url, "", params, headers);
    System.out.println(result);
}
```

Sample code (C#) *示例代码（C#）*

```csharp
using System;
using System.Net.Http;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace Withdraw
{
    class Program
    {
        static void Main(string[] args)
        {
            Dictionary<string, string> parameters = new Dictionary<string, string>();
            parameters.Add("loginId", "PA10000000");
            parameters.Add("amount", "10");

            Dictionary<string, string> headers = new Dictionary<string, string>();
            headers.Add("userCode","CO1AP1");
            headers.Add("token","DIwYQJZIYiosWrbXQe+TFdyMk6POZvbcM1KjxxQObZEn0+efzoM
Cb3i+PWr1ZFuj0UciR8w+qqo1M2hJ965Y9w==");

            Console.WriteLine(HttpUtils.PostRequest("http://paapiqat.oreo88.com/b2b/
player/withdraw", parameters, headers).Result);
            Console.ReadKey();
        }
    }
}
```

Response 反应

Return Good Case

Data will return JSON format.

数据将返回 JSON 格式

```
{
  'userCode': 'PA10000000',     // string
  'loginId': 'PA10.0',     // string
  'availableBalance': 1010.00,    // decimal(after withdrawal)
  'amount': 10.00      // decimal
}
```

3.9 FP007 – Change Status Member 更改用户状态

Update status for player.

更新用户状态

Endpoint

Name	Value	Description
URL	/player/update-status	
Method	GET/POST	

Headers 头文件

Name	Value	Validation	Description
userCode	String (required 必需项)		This is agent code that you get at step 2. E.g: CO1AP1 此为在第二步骤获取的代理编号，例如，CO1AP1
token	String (required 必需项)	*Token is available in **15 minutes** since it was created* *令牌在创建之后的 15 分钟内有效*	

Parameters 参数

Name	Value	Validation	Description
userCode	String (required 必需项)		This is user code / loginID of player. E.g: PA10000000 此为玩家登录名/用户名，例如 PA10000000
status	String (required 必需项)	ACTIVE, INACTIVE, SUSPENDED	Appendix User Status

Note:
- ‘**INACTIVE**’ player **CANNOT** login on Member site
- ‘**SUSPENDED**’ player **CAN** login on Member site, but **CANNOT** place bet

注意：
- ‘**INACTIVE**’ 玩家不可登陆
- ‘**SUSPENDED**’ 玩家可以登陆，但不能投注

Sample code (java) - See HttpUtils class at Appendix

示例代码（java）– 请参阅附录里的 HttpUtils class

```java
import java.io.IOException;
import java.util.HashMap;
import java.util.Map;

private static void updateStatusMember() throws IOException {
    Map params = new HashMap();
    params.put("userCode", "PA10000000");
    params.put("status", "ACTIVE");

    Map headers = new HashMap();
    headers.put("userCode", "CO1AP1");
    headers.put("token", "gvbLZb70DhMNXMLw3egvktPT3sfDWnDC5QI97MXHqK9FCFO1n6oepaXQivCIVpsDlth/jbR/qgUcpi2wHc62Tw==");

    String url = "http://paapiqat.oreo88.com/b2b/player/update-status";
    String result = HttpUtils.post(url, "", params, headers);
    System.out.println(result);
}
```

Sample code (C#) 示例代码（C#）

```csharp
using System;
using System.Net.Http;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace ChangeStatusMember
{
     class Program
```

```
    {
        static void Main(string[] args)
        {
            Dictionary<string, string> parameters = new Dictionary<string, string>()
;
            parameters.Add("userCode", "PA10000000");
            parameters.Add("status", "ACTIVE");

            Dictionary<string, string> headers = new Dictionary<string, string>();
            headers.Add("userCode","CO1AP1");
            headers.Add("token","DIwYQJZIYiosWrbXQe+TFdyMk6POZvbcM1KjxxQObZEn0+efzoM
Cb3i+PWr1ZFuj0UciR8w+qqo1M2hJ965Y9w==");

            Console.WriteLine(HttpUtils.PostRequest("http://paapiqat.oreo88.com/b2b/
player/update-status", parameters, headers).Result);
            Console.ReadKey();
        }
    }
}
```

Response 回应

```
{
"status": "successful"
}
```

3.10 FR001 – Wagers 投注

Get all your players wagers.

获取你用户所有的投注

*We also provide wagers feed to Push wagers changes to B2B customer servers (*see FR004*)
我们也提供投注信息来推动更改的投注到 B2B 客户的服务器（参考 FR004）

General 通用

Name	Value	Description
URL	/report/wagers	
Method	GET	

Headers 头文件

Name	Value	Validation	Description
userCode	String (required 必需项)		This is agent code that you get at step 2. E.g: CO1AP1 此为在第二步骤获取的代理编号，例如，CO1AP1

For internal use only

token	String (required 必需项)	*Token is available in **15 minutes** since it was created* *令牌在创建之后的 15 分钟内有效*	

Parameters 参数

Name	Value	Validation	Description
dateFrom	Date (required 必需项)	Created Date Time format yyyy-MM-dd HH:mm:ss GMT-4	*Example: 2016-10-15 23:59:59*
dateTo	Date (required 必需项)	Created Date Time format yyyy-MM-dd HH:mm:ss GMT-4	*Example: 2f016-10-16 23:59:59* *Rule: dateTo – dateFrom <= 24 hours*
product	String (required 必需项)	SB	Product Sport Book 产品: 体育
userCode	String (required 必需项)		This is user code / loginID of player. E.g: PA10000000 此为玩家登录名/用户名，例如 PA10000000
settle	Boolean (optional 非必需项)	true/false *(Default: false)*	If true is wager status include: SETTLED, CANCELLED, DELETED Else include: OPEN, PENDING
filterBy	String (optional 非必需项)	event_date/wager_date *(Default: event_date)*	In case settle equals FALSE is date range base on filterBy's value ELSE based on settlement date 如果 settle 返回 FAISE, 日期会按照

			filterBy 的数值，否则按照注单结算日期
locale	String (optional 非必需项)	List supported locales based on available brand’s languages.	Appendix Locale (Language)

Sample code (java) - *See HttpUtils class at Appendix*

示例代码（java）– 请参阅附录里的 HttpUtils class

```java
import java.io.IOException;
import java.util.HashMap;
import java.util.Map;

private static void getListWager() throws IOException {
    Map params = new HashMap();
    params.put("userCode", "PA10000000");
    params.put("dateFrom", "2017-04-13 00:00:00");
    params.put("dateTo", "2017-04-14 00:00:00");
    params.put("settle", "true");
    params.put("product", "SB");
    Map headers = new HashMap();
    headers.put("userCode", "CO1AP1");
    headers.put("token", "gvbLZb70DhMNXMLw3egvktPT3sfDWnDC5QI97MXHqK9FCFO1n6oepaXQivCIVpsDlth/jbR/qgUcpi2wHc62Tw==");

    String url = "http://paapiqat.oreo88.com/b2b/report/wagers";
    String result = HttpUtils.get(url, "", params, headers);
    System.out.println(result);
}
```

Sample code (C#) 示例代码（C#）

```csharp
using System;
using System.Net.Http;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace Wagers
{
    class Program
    {
        static void Main(string[] args)
        {
            Dictionary<string, string> parameters = new Dictionary<string, string>();
            parameters.Add("userCode", "PA10000000");
            parameters.Add("dateFrom", "2017-04-13 00:00:00");
            parameters.Add("dateTo", "2017-04-14 00:00:00");
            parameters.Add("settle", "true");
            parameters.Add("product", "SB");

            Dictionary<string, string> headers = new Dictionary<string, string>();
            headers.Add("userCode","CO1AP1");
            headers.Add("token","DIwYQJZIYiosWrbXQe+TFdyMk6POZvbcM1KjxxQObZEn0+efzoMCb3i+PWr1ZFuj0UciR8w+qqo1M2hJ965Y9w==");

            Console.WriteLine(HttpUtils.GetRequest("http://paapiqat.oreo88.com/b2b/report/wagers", parameters, headers).Result);
            Console.ReadKey();
        }
    }
}
```

Response OK (SB - without Parlay) 反应 OK (体育-没有过关)

```
[
  {
    "wagerId": 6719824,
    "eventName": "Brisbane Olympic United SSC-vs-Moreton Bay United FC",
    "parentEventName": null,
    "headToHead": null,
    "wagerDateFm": "2017-04-13 05:13:04",
    "eventDateFm": "2017-04-13",
    "settleDateFm": "2017-04-13 08:12:05",
    "status": "SETTLED",
    "homeTeam": "Brisbane Olympic United SSC",
    "awayTeam": "Moreton Bay United FC",
    "selection": "Moreton Bay United FC",
    "handicap": 0.75,
    "odds": 1.99,
    "oddsFormat": 1,
    "betType": 2,
    "league": "Australia NPL - Queensland",
    "stake": 7,
    "sport": "Soccer",
    "currencyCode": "CNY",
    "inplayScore": "0-1",  //the live event's current score
    "inPlay": true,  //indicate that the wager is placed on In Play event
    "homePitcher": null,
    "awayPitcher": null,
    "homePitcherName": null,
    "awayPitcherName": null,
    "period": 0,
    "cancellationStatus": null,
    "parlaySelections": [],
    "category": null,
    "toWin": 6.93,  //is the amount that player will win if he wins the best
    "toRisk": 7,  //is the amount that player will lose if he loses the bet
    "product": "SB",
    "parlayMixOdds": 1.99,
    "competitors": [],
    "userCode": "Q23100000D",
    "loginId": "Q231.0D",
    "winLoss": -3.5,
    "turnover": 3.5,
    "commission": 0,
    "scores": [
      {
        "period": 1,
        "score": "0-1"
      },
      {
        "period": 0,
        "score": "2-2"
      }
    ],
    "result": "LOSE"
  }
]
```

Response OK (SB - with Parlay) 反应OK (体育 – 有过关)

```
[
  {
    "wagerId": 6719844,
    "eventName": null,
    "parentEventName": null,
    "headToHead": null,
    "wagerDateFm": "2017-04-13 05:19:37",
    "eventDateFm": "2017-04-13",
    "settleDateFm": "2017-04-13 06:22:14",
    "status": "SETTLED",
    "homeTeam": null,
    "awayTeam": null,
    "selection": "",
    "handicap": 0,
    "odds": 5.597,
    "oddsFormat": 1,
    "betType": 6,
    "league": null,
    "stake": 7,
    "sport": null,
    "currencyCode": "CNY",
    "inplayScore": "0",  //the live event's current score
    "inPlay": false,  //indicate that the wager is placed on In Play event
    "homePitcher": null,
    "awayPitcher": null,
    "homePitcherName": null,
    "awayPitcherName": null,
    "period": 0,
    "cancellationStatus": null,
    "category": null,
    "parlaySelections": [
      {
        "selection": "Banyule City",
        "eventDateFm": "2017-04-13",
        "scores": [
          {
            "period": 1,
            "score": "1-1"
          },
          {
            "period": 0,
            "score": "1-1"
          }
        ],
        "sport": "Soccer",
        "league": "Australia - FFA Cup Qualifiers",
        "eventName": "Banyule City -vs- Melbourne Knights",
        "homeTeam": "Banyule City",
        "awayTeam": "Melbourne Knights",
        "betType": 2,
        "wagerId": 6719844,
        "inplayScore": null,  //the live event's current score
        "inPlay": false,  //indicate that the wager is placed on In Play event
        "odds": 1.952,
        "handicap": 2.25,
        "homePitcher": null,
        "awayPitcher": null,
        "homePitcherName": null,
        "awayPitcherName": null,
        "period": 0,
        "legStatus": "WON"
      },
      {
        "selection": "Brisbane Wolves",
        "eventDateFm": "2017-04-13",
        "scores": [
          {
            "period": 1,
            "score": "0-1"
          },
          {
            "period": 0,
            "score": "0-4"
          }
        ],
        "sport": "Soccer",
        "league": "Australia - FFA Cup Qualifiers",
        "eventName": "Brisbane Wolves -vs- Peninsula Power",
        "homeTeam": "Brisbane Wolves",
        "awayTeam": "Peninsula Power",
        "betType": 2,
        "wagerId": 6719844,
        "inplayScore": null,  //the live event's current score
        "inPlay": false,  //indicate that the wager is placed on In Play event
        "odds": 1.769,
        "handicap": 1.25,
        "homePitcher": null,
        "awayPitcher": null,
        "homePitcherName": null,
        "awayPitcherName": null,
        "period": 0,
        "legStatus": "LOSE"
      },
      {
        "selection": "AZS UJ Krakow (women)",
        "eventDateFm": "2017-04-13",
        "scores": [
          {
            "period": 0,
            "score": "4-1"
          }
        ],
        "sport": "Soccer",
        "league": "Poland - Ekstraliga Women",
        "eventName": "Gks Gornik Leczna (W) -vs- AZS UJ Krakow (women)",
        "homeTeam": "Gks Gornik Leczna (W)",
        "awayTeam": "AZS UJ Krakow (women)",
        "betType": 2,
        "wagerId": 6719844,
        "inplayScore": null,  //the live event's current score
        "inPlay": false,  //indicate that the wager is placed on In Play event
        "odds": 1.621,
        "handicap": 4.5,
        "homePitcher": null,
        "awayPitcher": null,
        "homePitcherName": null,
        "awayPitcherName": null,
        "period": 0,
        "legStatus": "WON"
      }
    ],
    "toWin": 32.179,  //is the amount that player will win if he wins the best
    "toRisk": 7, //is the amount that player will lose if he loses the best
    "product": "SB",
    "parlayMixOdds": 5.597,
    "competitors": [],
    "userCode": "Q23100000D",
    "loginId": "Q231.0D",
    "winLoss": -7,
    "commission": 0,
    "scores": [],
    "turnover": 7,
    "result": "LOSE"
  }
]
```

Response OK (SB - with Outright)反应OK (体育 – 有优胜冠军)

```
[
  {
    "wagerId": 6862959,
    "eventName": "Both Teams To Score 1st Half",
    "parentEventName": "Middlesbrough-vs-Arsenal",
    "headToHead": null,
    "wagerDateFm": "2017-04-17 07:15:42",
    "eventDateFm": "2017-04-17",
    "settleDateFm": "2017-04-17 08:50:00",
    "status": "SETTLED",
    "homeTeam": "No",
    "awayTeam": "Both Teams To Score 1st Half",
    "selection": "No",
    "handicap": 0,
    "odds": 1.199,
    "oddsFormat": 1,
    "betType": 99,
    "league": "England - Premier League",
    "stake": 10,
    "sport": "Soccer",
    "currencyCode": "CNY",
    "inplayScore": "",  //the live event's current score
    "inPlay": false,  //indicate that the wager is placed on In Play event
    "homePitcher": null,
    "awayPitcher": null,
    "homePitcherName": null,
    "awayPitcherName": null,
    "period": 0,
    "cancellationStatus": null,
    "parlaySelections": [],
    "category": "Both Teams To Score ",
    "toWin": 1.99,  //is the amount that player will win if he wins the best
    "toRisk": 10,  //is the amount that player will lose if he loses the best
    "product": "SB",
    "parlayMixOdds": 1.199,
    "competitors": [],
    "userCode": "Q23100000D",
    "loginId": "Q231.0D",
    "winLoss": 1.99,
    "commission": 0.002,
    "scores": [],
    "turnover": 10,
    "result": "WIN"
  }
]
```

Response OK (SB - with Teaser) 反应OK (体育-含变让分过关盘)

```
[
  {
    "wagerId": 702256132,
    "eventName": null,
    "parentEventName": null,
    "headToHead": null,
    "wagerDateFm": "2018-05-02 04:15:25",
    "eventDateFm": "2018-05-02",
    "settleDateFm": null,
    "status": "OPEN",
    "homeTeam": null,
    "awayTeam": null,
    "selection": null,
    "handicap": 0,
    "odds": 0.5,
    "oddsFormat": 2,
    "betType": 7,
    "league": null,
    "stake": 11,
    "sport": null,
    "currencyCode": "CNY",
    "inplayScore": "0",
    "inPlay": false,
    "homePitcher": null,
    "awayPitcher": null,
    "homePitcherName": null,
    "awayPitcherName": null,
    "period": 0,
    "cancellationStatus": null,
    "parlaySelections": [
      {
        "selection": "Toronto Raptors",
        "eventDateFm": "2018-05-04",
        "scores": [],
        "sport": "Basketball",
        "league": "NBA",
        "homeTeam": "Toronto Raptors",
        "awayTeam": "Cleveland Cavaliers",
        "betType": 2,
        "wagerId": 702256132,
        "selectionType": 0,
        "inplayScore": null,
        "inPlay": false,
        "odds": 0,
        "handicap": 0.5,
        "homePitcher": null,
        "awayPitcher": null,
        "homePitcherName": null,
        "awayPitcherName": null,
        "period": 0,
        "legStatus": "ACCEPTED",
        "eventName": "Toronto Raptors-vs-Cleveland Cavaliers",
        "point": 7,
        "type": "NBA 2 - 6 Team",
        "teaserHDPPoint": "-6.5 + 7 pts"
      },
      {
        "selection": "Utah Jazz",
        "eventDateFm": "2018-05-03",
        "scores": [],
        "sport": "Basketball",
        "league": "NBA",
        "homeTeam": "Houston Rockets",
        "awayTeam": "Utah Jazz",
        "betType": 2,
            "wagerId": 702256132,
            "selectionType": 1,
            "inplayScore": null,
            "inPlay": false,
            "odds": 0,
            "handicap": 18,
            "homePitcher": null,
            "awayPitcher": null,
            "homePitcherName": null,
            "awayPitcherName": null,
            "period": 0,
            "legStatus": "WON",
            "eventName": "Houston Rockets-vs-Utah Jazz",
            "point": 7,
            "type": "NBA 2 - 6 Team",
            "teaserHDPPoint": "11 + 7 pts"
          }
        ],
        "category": null,
        "toWin": 5.5,
        "toRisk": 11,
        "product": "SB",
        "parlayMixOdds": 0.5,
        "competitors": [],
        "userCode": "Q231000000",
        "loginId": "Q231000000",
        "winLoss": 0,
        "turnover": 0,
        "scores": [],
        "result": null
    }
]
```

3.11 FR002 – All Wagers FR002 – 所有投注

Get all your players wagers (this function work as FR001 and replace it in the future).

获取你用户所有的投注 （此功能和 FR001 一样，并在将来取代它）

General

Name	Value	Description
URL	/report/all-wagers	
Method	GET	

Headers

Name	Value	Validation	Description
userCode	String (required 必需项)		This is agent code that you get at step 2. E.g: CO1AP1 此为在第二步骤获取的代理编号，例如，CO1AP1

token	String (required 必需项)	*Token is available in **15 minutes** since it was created 令牌在创建之后的 15 分钟内有效*	

Parameters

Name	Value	Validation	Description
dateFrom	Date (optional 非必需项)	Created Date Time format yyyy-MM-dd HH:mm:ss GMT-4	*Example: 2016-10-15 23:59:59*
dateTo	Date (optional 非必需项)	Created Date Time format yyyy-MM-dd HH:mm:ss GMT-4	*Example: 2016-10-16 23:59:59* *Rule: dateTo – dateFrom <= 24 hours* *限制: dateTo – dateFrom <= 24 小时*
Note: 1. WITHOUT date range: a. System shall response all wagers from last 24 hours. 2. Specific date range: a. If **userCode**[1] = null, valid date range will be up to 24 hours. b. If **userCode**[1] != null, valid date range will be up to 168 hours (7 days). **注意:** 1. 如未输入日期范围, 则系统应返回过去 24 小时内的全部注单。 2. 如果输入了日期范围: a. 如果 **userCode**[1] = null, 有效日期范围最高为 24 小时。 b. 如果 **userCode**[1] != null, 有效日期范围最高为 24 小时(7 天).			
userCode[1]	String (optional 非必需项)		This is user code / loginID of player 玩家用户名/登录名
settle	Int (optional 非必需项)	*1: settle* *0: unsettle* *-1: all (both settle and unsettle)* *(Default: -1)*	If 1 is wager status include: SETTLED, CANCELLED, DELETED Else if 0 include: OPEN, PENDING Else all status
filterBy	String (optional 非必需项)	event_date wager_date	

		settle_date update_date *(Default:* *wager_date)*	
Note: **In case filterBy is settle_date is system only query data by date format but dateFrom and dateTo still use format yyyy-MM-dd HH:00:00**			
locale	String (optional 非必需项)	List supported locales based on available brand’s languages.	Appendix Locale (Language)
wagerIds	String (optional 非必需项)	This is list of wager id, specified as the comma-separated	Example: **6862955,6862947**
URL example: http://paapiqat.oreo88.com/b2b/report/all-wagers?wagerIds=6862955,6862947			

Response OK 返回 OK

The response result as same with FR001 but add more “userCode” per wager result.
这个反应是跟 FR001 一样，但会为每张注单结果添加“userCode”

```
[
  {
    "wagerId": 6862955,
    "eventName": "Winner of 2018 Super Bowl?",
    "parentEventName": null,
    "headToHead": null,
    "wagerDateFm": "2017-04-17 07:15:29",
    "eventDateFm": "2017-09-01",
    "settleDateFm": null,  // in case status=”SETTLED” is value like "2017-09-01 22:02:15”
    "status": "OPEN",
    "homeTeam": "New England Patriots",
    "awayTeam": "Winner of 2018 Super Bowl?",
    "selection": "New England Patriots",
    "handicap": 0,
    "odds": 5.39,
    "oddsFormat": 1,
    "betType": 99,
    "league": "NFL",
    "stake": 10,
    "sport": "Football",
    "currencyCode": "CNY",
    "inplayScore": "",  //the live event's current score
    "inPlay": false,  //indicate that the wager is placed on In Play event
    "homePitcher": null,
    "awayPitcher": null,
    "homePitcherName": null,
    "awayPitcherName": null,
    "period": 0,
    "cancellationStatus": null,
    "parlaySelections": [],
    "category": "Futures",
    "toWin": 43.9,  //is the amount that player will win if he wins the best
    "toRisk": 10,  //is the amount that player will lose if he loses the best
    "product": "SB",
    "parlayMixOdds": 5.39,
    "competitors": [],
    "userCode": "Q23100000D",
    "loginId": "Q231.0D",
    "winLoss": 0,
    "commission": 0,
    "scores": [],
    "result": null
  },
  {
    "wagerId": 6862947,
    "eventName": "Player to Win ATP French Open ? (All In)",
    "parentEventName": null,
    "headToHead": null,
    "wagerDateFm": "2017-04-17 07:11:17",
    "eventDateFm": "2017-05-22",
    "settleDateFm": null,
    "status": "OPEN",
    "homeTeam": "Andy Murray",
    "awayTeam": "Player to Win ATP French Open ? (All In)",
    "selection": "Andy Murray",
    "handicap": 0,
    "odds": 5.36,
    "oddsFormat": 1,
    "betType": 99,
    "league": "ATP French Open",
    "stake": 10,
    "sport": "Tennis",
    "currencyCode": "CNY",
    "inplayScore": "",  //the live event's current score
    "inPlay": false,  //indicate that the wager is placed on In Play event
    "homePitcher": null,
    "awayPitcher": null,
    "homePitcherName": null,
    "awayPitcherName": null,
    "period": 0,
    "cancellationStatus": null,
    "parlaySelections": [],
    "category": "To Win",
    "toWin": 43.6, //is the amount that player will win if he wins the best
    "toRisk": 10,  //is the amount that player will lose if he loses the best
    "product": "SB",
    "parlayMixOdds": 5.36,
    "competitors": [],
    "userCode": "Q23100000D",
    "loginId": "Q231.0D",
    "winLoss": 0,
    "commission": 0,
    "scores": [],
    "turnover": 0,
    "result": null
  }
]
```

3.12 FR003 – Win Loss Simple FR003 - 简易盈亏

Get win loss simple of agent and player

获取代理和用户的简易盈亏

General 通用

Name	Value	Description

URL	/report/winloss-simple	
Method	GET	

Headers 头文件

Name	Value	Validation	Description
userCode	String (required 必需项)		This is agent code that you get at step 2. E.g: CO1AP1 此为在第二步骤获取的代理编号，例如，CO1AP1
token	String (required 必需项)	*Token is available in **15 minutes** since it was created 令牌在创建之后的 15 分钟内有效*	

Parameters 参数

Name	Value	Validation	Description
dateFrom	Date (optional 非必需项)	Created Date format yyyy-MM-dd GMT-4	Date format *Example: 2016-10-15*
dateTo	Date (optional 非必需项)	Created Date format yyyy-MM-dd GMT-4	Date format *Example: 2016-10-16* *Rule: dateTo – dateFrom <= 90 days*
userCode[2]	String (optional 非必需项)		This is user code / loginID of **AGENT** or **PLAYER**

Note:

1. Date range = null:
   System get reports of TODAY.
2. **userCode[2]** = null
   Get reports of user all under userCode who calling API
3. **userCode[2]** != null (specific user code or login id)
   a. If **userCode[2]** is agent (level != **PL**), get reports of user all under **userCode[2]**.
   b. If **userCode[2]** is player (level = **PL**), get reports of player.

For example:

userCode[2] = CO1AP100 - This is user code of **Agent**
userCode[2] = PA10000000 - This is user code of **Player**

注意:

1. Date range = null:

系统会获取当日报表。

2. userCode[2] = null

   获取 userCode 下面所有调用 API 的用户报表。

3. userCode[2] != null (特定用户名或登录名)

   a. 如果 userCode[2] 为代理(level != **PL**), 获取 userCode[2] 下所有用户的报表。

   b. 如果 userCode[2] 为玩家(level = **PL**), 获取玩家报表。

For example:

userCode[2] = CO1AP100 - 此为代理用户名。

userCode[2] = PA10000000 – 此为玩家用户名。

```java
import java.io.IOException;
import java.util.HashMap;
import java.util.Map;

private static void getWinLossSimple() throws IOException {
    Map params = new HashMap();
    params.put("userCode", "PA10000000");  // params.put("userCode", "CO1AP100");
    params.put("dateFrom", "2017-05-01");
    params.put("dateTo", "2017-06-05");
    Map headers = new HashMap();
    headers.put("userCode", "CO1AP1");
    headers.put("token", "gvbLZb70DhMNXMLw3egvktPT3sfDWnDC5QI97MXHqK9FCFO1n6oepaXQivCIVpsDlth/jbR/qgUcpi2wHc62Tw==");

    String url = "http://paapiqat.oreo88.com/b2b/report/winloss-simple";
    String result = HttpUtils.get(url, "", params, headers);
    System.out.println(result);
}
```

Sample code (C#) 示例代码（C#）

```csharp
using System;
using System.Net.Http;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace WinLostSimple
{
    class Program
    {
        static void Main(string[] args)
        {
            Dictionary<string, string> parameters = new Dictionary<string, string>();
            parameters.Add("userCode", "PA10000000");
            parameters.Add("dateFrom", "2017-05-01");
            parameters.Add("dateTo", "2017-06-05");

            Dictionary<string, string> headers = new Dictionary<string, string>();
            headers.Add("userCode","CO1AP1");
            headers.Add("token","DIwYQJZIYiosWrbXQe+TFdyMk6POZvbcM1KjxxQObZEn0+efzoMCb3i+PWr1ZFuj0UciR8w+qqo1M2hJ965Y9w==");

            Console.WriteLine(HttpUtils.GetRequest("http://paapiqat.oreo88.com/b2b/report/winloss-simple", parameters, headers).Result);
            Console.ReadKey();
        }
    }
}
```

Response OK 返回 OK

```
{
  "report": [
    {
      "userId": 1000002738,
      "userCode": "PA10000000",
      "firstName": "",
      "currency": "USD",
      "levelName": "PL",
      "loginId": "PA1.000",
      "turnover": 19,
      "grossComm": 0.24,
      "payout": -3.02,
      "winLoss": 0,
      "comm": 0.22,
      "volume": 18.51,
      "total": 0.22,
      "upline": 2.8
    },
    {
      "userId": 1000002796,
      "userCode": "PA10000001",
      "firstName": "",
      "currency": "USD",
      "levelName": "PL",
      "loginId": " PA1.001",
      "turnover": 1,
      "grossComm": 0.01,
      "payout": -1,
      "winLoss": 0.05,
      "comm": 0,
      "volume": 1,
      "total": 0.05,
      "upline": 0.95
    }
  ],
  "total": {
    "turnover": 20,
    "grossComm": 0.25,
    "payout": -4.02,
    "winLoss": 0.05,
    "comm": 0.22,
    "volume": 19.51,
    "total": 0.27,
    "upline": 3.74
  }
}
```

3.13 FR004 – GET MY BET FR004 – 获取我的投注

This service is used generate URL by pass login for player then go to my bet page.

此服务透过传送登陆产生 URL 给用户到‘我的投注’页面。

Endpoint

Name	Value	Description
URL	/player/account/my-bets-full	

Method	GET/POST	

Headers 头文件

Name	Value	Validation	Description
userCode	String (required 必需项)		This is agent code that you get at step 2. E.g: CO1AP1 此为在第二步骤获取的代理编号，例如，CO1AP1
token	String (required 必需项)	*Token is available in **15 minutes** since it was created 令牌在创建之后的 15 分钟内有效*	

Parameters 参数

Name	Value	Validation	Description
loginId	String (required 必需项)		This is user code / loginID of player. E.g: PA10000000 or PA10.02 此为玩家登录名/用户名，例如 PA10000000 或 PA10.abc123
locale	String (optional 非必需项)	List supported locales based on available brand's languages.	Appendix Locale (Language)

Format URL Go to My Bet 前往"我的投注" 的 URL 格式

<host>/member-service/v1/account/my-bets-full?locale=<en>&token=<token>

Example:

http://wlqat.oreo88.com/member-service/v1/account/my-bets-full?locale=en&token=YkZXSFl5S0VsUElrQzRlRS9ZRERsbDN3dktnb2dnSE1tcTZ1VTEvVnFldTJkVTcyWklJeVVRSk1xY3dpM2VWVXFYY3lxMzRpWGNBWllsakV2Wk1JZ0E9PQ==

Note: this token generated by our system for player login to our system. It is not token that 3rd party generate.

注意: 这个代码是由平博系统产生让用户登入系统。 这个不是第三方产生的代码。

Sample code (java) - *See HttpUtils class at Appendix*

示例代码（java） – 请参阅附录里的 HttpUtils class

```
import java.io.IOException;
```

```java
import java.util.HashMap;
import java.util.Map;
private static void getTokenLogin() throws IOException {
    Map params = new HashMap();
    params.put("loginId", "PA10.02");
    params.put("locale", "zh-cn");
    Map headers = new HashMap();
    headers.put("userCode", "CO1AP1");
    headers.put("token", "gvbLZb70DhMNXMLw3egvktPT3sfDWnDC5QI97MXHqK9FCFO1n6oepaXQivCIVpsDlth/jbR/qgUcpi2wHc62Tw==");
    String url = "http://paapiqat.oreo88.com/b2b/player/account/my-bets-full";
    String result = HttpUtils.post(url, "", params, headers);
    System.out.println(result);
}
```

Sample code (C#) 示例代码（C#）

```csharp
using System;
using System.Net.Http;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace GetMyBet
{
    class Program
    {
        static void Main(string[] args)
        {
            Dictionary<string, string> parameters = new Dictionary<string, string>();
            parameters.Add("loginId", "PA10.02");
            parameters.Add("locale", "zh-cn");

            Dictionary<string, string> headers = new Dictionary<string, string>();
            headers.Add("userCode","CO1AP1");
            headers.Add("token","DIwYQJZIYiosWrbXQe+TFdyMk6POZvbcM1KjxxQObZEn0+efzoMCb3i+PWr1ZFuj0UciR8w+qqo1M2hJ965Y9w==");

            Console.WriteLine(HttpUtils.PostRequest("http://paapiqat.oreo88.com/b2b/player/account/my-bets-full", parameters, headers).Result);
            Console.ReadKey();
        }
    }
}
```

Response 反应

Result is URL to login System. Then we use this url to open new popup in browser. This url will auto login Pinbet88 system and redirect to My Bet page.

反应结果是透过 URL **登入系**统。然后我们使用这个 URL **在**浏览器中打开新的弹出窗口。这个 URL **将自**动登入 Pinbet88 **系**统并重新引导至"**我的投注**"页面。

```json
{
  "userCode": "PA10000001",
  "loginId": "PA10.02",
  "token": "UGd0eFhVaUlpTXpRelhZN001NG8wSVR0eW5xMkVWcGpNUXlJYVdrUTJid2p1YW9BdzBIMGh0NmRZRG9KL3B4TXlyMjk1SjBGMnU2NzRwcVdmYXpYakE9PQ==",
  "loginUrl": "http://wlqat.oreo88.com/member-service/v1/account/my-bets-full?locale=zh-cn&token=UGd0eFhVaUlpTXpRelhZN001NG8wSVR0eW5xMkVWcGpNUXlJYVdrUTJid2p1YW9BdzBIMGh0NmRZRG9KL3B4TXlyMjk1SjBGMnU2NzRwcVdmYXpYakE9PQ==",
    "updateDate": "2017-09-06 22:46:34"
}
```

3.14 FR005 – Wager Feed FR005 – 投注资料

This service will push wagers changes to B2B customer servers via HTTP

这项服务通过 HTTP 推动投注更改信息到 B2B 客户的服务器。

To use this function, B2B customer server need to public one Restful service to receive JSON data from us. After B2B customer config success, you need to send back to us your's public URL

为了能使用这个功能，B2B 客户服务器需要公开一个 Restful 服务来接收来自我们的 JSON 数据。在 B2B 客户配置成功后，您需要将您的公开网址发回给我们

Request URL: url_any

Request Method: POST

Request Payload: (*see define message model for detail*)

Message model:

```
{
"messageId":123456789,
"messageData":[
{feed_wager_data1}, {feed_wager_data2}
]
}
```

Feed_wager_data model: (this model is same with FR002's model except id field added more

此数据模型与 FR002 内的模型一样，除了 id 项添加了更多)

```
{
  "id": 750,
  "wagerId": 7838345,
  "sport": null,
  "league": null,
  "eventName": null,
  "parentEventName": null,
  "headToHead": null,
  "scores": [],
  "homeTeam": null,
  "awayTeam": null,
  "selection": null,
  "parlaySelections": [
    {
      "wagerId": 7838345,
      "selection": "Over",
      "eventDateFm": "2017-07-26",
      "scores": [
        {
          "period": 1,
          "score": "0-2"
        },
        {
          "period": 0,
          "score": "0-3"
        }
      ],
      "sport": "Soccer",
      "league": "UEFA - EURO Women",
      "homeTeam": "Iceland (W)",
      "awayTeam": "Austria (W)",
      "odds": 1.84,
      "handicap": 0.75,
      "legStatus": "WON",
      "homePitcher": null,
      "awayPitcher": null,
      "homePitcherName": null,
      "awayPitcherName": null,
      "betType": 3,
      "inplayScore": null,
      "inPlay": false,
      "period": 1,
      "selectionType": 3,
      "eventname": "Iceland (W)-vs-Austria (W)"

    },
    {
      "wagerId": 7838345,
      "selection": "Over",
      "eventDateFm": "2017-07-26",
      "scores": [
        {
          "period": 1,
          "score": "2-0"
        },
        {
          "period": 0,
          "score": "3-1"
        }
      ],
      "sport": "Soccer",
      "league": "UEFA - Champions League Qualifiers",
      "homeTeam": "FC Astana",
      "awayTeam": "Legia Warszawa",
      "odds": 2,
      "handicap": 2.25,
      "legStatus": "WON",
      "homePitcher": null,
      "awayPitcher": null,
      "homePitcherName": null,
      "awayPitcherName": null,
      "betType": 3,
      "inplayScore": null,
      "inPlay": false,
      "period": 0,
      "selectionType": 3
      "eventname": " FC Astana-vs-Legia Warszawa"
    },
    {
      "wagerId": 7838345,
      "selection": "Red Bull Salzburg",
      "eventDateFm": "2017-07-26",
      "scores": [
        {
          "period": 1,
          "score": "0-1"
```

```
     },
     {
      "period": 0,
      "score": "1-1"
     }
    ],
    "sport": "Soccer",
    "league": "UEFA - Champions League Qualifiers",
    "homeTeam": "Red Bull Salzburg",
    "awayTeam": "HNK Rijeka",
    "odds": 1.636,
    "handicap": 0,
    "legStatus": "LOSE",
        "homePitcher": null,
        "awayPitcher": null,
        "homePitcherName": null,
        "awayPitcherName": null,
        "betType": 1,
        "inplayScore": null,
        "inPlay": false,
        "period": 0,
        "selectionType": 0
       },
       {
        "wagerId": 7838345,
        "selection": "Under",
        "eventDateFm": "2017-07-26",
        "scores": [
         {
          "period": 1,
          "score": "0-0"
         },
         {
          "period": 0,
          "score": "1-0"
         }
        ],
        "sport": "Soccer",
        "league": "UEFA - Champions League Qualifiers",
        "homeTeam": "Viitorul Constanta",
        "awayTeam": "APOEL Nicosia",
        "odds": 1.657,
        "handicap": 2.25,
        "legStatus": "WON",
        "homePitcher": null,
        "awayPitcher": null,
        "homePitcherName": null,
        "awayPitcherName": null,
        "betType": 3,
        "inplayScore": null,
        "inPlay": false,
        "period": 0,
        "selectionType": 4
        "eventname": "Viitorul Constanta-vs-APOEL Nicosia"
       }
      ],
      "odds": 9.976,
      "handicap": 0,
      "parlayMixOdds": 9.976,
      "homePitcher": null,
      "awayPitcher": null,
      "homePitcherName": null,
      "awayPitcherName": null,
      "betType": 6,
      "inplayScore": "0",
      "cancellationStatus": null,
      "category": null,
      "inPlay": false,
      "eventDateFm": "2017-07-26",
      "oddsFormat": 1,
      "result": "DRAW",
      "status": "SETTLED",
      "toWin": 89.76,
      "toRisk": 10,
      "stake": 10,
      "period": 0,
      "product": "SB",
      "winLoss": 0,
      "currencyCode": "CNY",
      "exRate": 0.1480451,
      "userCode": "KR00000000",
      "loginId": "KR0.000",
      "wagerDateFm": "2017-07-26 02:13:31"
      "turnover": 0,
      "settleDateFm":  "2017-07-26 04:11:22"
    }
```

Example, Coding: 例如，代码：

Setup endpoint 设置端点:

```
@RequestMapping(value = "/feed/wager", method = RequestMethod.POST)
public Object receiveData1(@RequestBody Message message) {
    System.out.println("======IntegrationController /feed/wager ========" + message.getMessageData().toString());
    String result = message.getMessageId();
    fService.onMessage(message.getMessageData());
    return result;
}
```

Define message model 界定信息模型:

```
public class Message {

    String messageId;
    List<WagerFeed> messageData;

    public String getMessageId() {
        return messageId;
    }

    public List<WagerFeed> getMessageData() {
        return messageData;
    }

    public void setMessageData(List<WagerFeed> messageData) {
        this.messageData = messageData;
    }
}
public class WagerFeed {

    long id;
    long wagerId;
    String status;
    BigDecimal toWin;
    BigDecimal toRisk;
    BigDecimal winLoss;
```

```java
    String currency;
    BigDecimal exRate;
    String userCode;
    String wagerDateFm;
    String sport;
    String league;
    String eventName;
    String parentEventName;
    String headToHead;
    List<EventScore> scores = new ArrayList<>();
    String homeTeam;
    String awayTeam;
    String selection;
    List<WagerFeedSelection> parlaySelections = new ArrayList<>();
    BigDecimal odds;
    BigDecimal handicap;
    BigDecimal parlayMixOdds;
    String homePitcher;
    String awayPitcher;
    String homePitcherName;
    String awayPitcherName;
    int betType;
    String inplayScore;
    String cancellationStatus;
    String category;
    boolean inPlay;
    String eventDateFm;
    int oddsFormat;
    String result;
    // GETTER SETTER HERE
}
public class EventScore {
    int period;
    String score;
    // GETTER SETTER HERE
}
public class WagerFeedSelection {
    long wagerId;
    String selection;
    String eventDateFm;
    List<EventScore> scores = new ArrayList<>();
    String sport;
    String league;
    String homeTeam;
    String awayTeam;
    BigDecimal odds;
    BigDecimal handicap;
    private LegStatus legStatus;
    String homePitcher;
    String awayPitcher;
    String homePitcherName;
    String awayPitcherName;
    int betType;
    String inplayScore;
    boolean inPlay;
    int period;
    int selectionType;
    String eventName;

    // GETTER SETTER HERE
}
```

3.15 FR006 – Announcement 公告

This service will get match announcement

这项服务用于获取比赛公告

General 通用

Name	Value	Description
URL	/player/account/announcements	
Method	GET, POST	

Headers 头文件

Name	Value	Validation	Description
userCode	String (required 必需项)		This is agent code that you get at step 2. E.g: CO1AP1
token	String (required 必需项)	*Token is available in **15 minutes** since it was created* *令牌在创建之后的 15 分钟内有效*	

Parameters 参数

Name	Value	Validation	Description
dateFrom	Date (optional 非必需项)	Created Date Time format yyyy-MM-dd HH:mm:ss GMT-4	*Example: 2016-10-15 23:59:59*
dateTo	Date (optional 非必需项)	Created Date Time format yyyy-MM-dd HH:mm:ss GMT-4	*Example: 2016-10-16 23:59:59* *Rule: dateTo – dateFrom <= 168 hours (7 days)*
Note: 1. **WITHOUT** date range: a. System shall response all announcement from last 24 hours (1 day). **注意:** 如果未输入日期，系统将返回过去 24 小时内的所有公告			
locale	String (optional 非必需项)	List supported locales based on available brand's languages.	Appendix Locale (Language)

sport	String (optional 非必需项)	The sport name	Appendix Sport

Sample code (java) - *See HttpUtils class at Appendix*

示例代码（java） – 请参阅附录里的 HttpUtils class

```java
import java.io.IOException;
import java.util.HashMap;
import java.util.Map;

private static void getAnnouncements() throws IOException {
    Map params = new HashMap();
    params.put("dateFrom", "2017-04-13 00:00:00");
    params.put("dateTo", "2017-04-14 00:00:00");
    params.put("locale", "en");
    params.put("sport", "soccer");
    Map headers = new HashMap();
    headers.put("userCode", "CO1AP1");
    headers.put("token", "gvbLZb70DhMNXMLw3egvktPT3sfDWnDC5QI97MXHqK9FCFO1n6oepaXQivCIVpsDlth/jbR/qgUcpi2wHc62Tw==");

    String url = "http://paapiqat.oreo88.com/b2b/player/account/announcements";
    String result = HttpUtils.get(url, "", params, headers);
    System.out.println(result);
}
```

Sample code (C#) 示例代码（C#）

```csharp
using System;
using System.Net.Http;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace Announcement
{
    class Program
    {
        static void Main(string[] args)
        {
            Dictionary<string, string> parameters = new Dictionary<string, string>();
            parameters.Add("dateFrom", "2017-04-13 00:00:00");
            parameters.Add("dateTo", "2017-04-14 00:00:00");
            parameters.Add("locale", "en");

            Dictionary<string, string> headers = new Dictionary<string, string>();
            headers.Add("userCode","CO1AP1");
            headers.Add("token","DIwYQJZIYiosWrbXQe+TFdyMk6POZvbcM1KjxxQObZEn0+efzoMCb3i+PWr1ZFuj0UciR8w+qqo1M2hJ965Y9w==");

            Console.WriteLine(HttpUtils.GetRequest("http://paapiqat.oreo88.com/b2b/player/account/announcements", parameters, headers).Result);
            Console.ReadKey();
        }
    }
}
```

Response OK 返回 OK

```
[
  {
    "text": "Wagers for the following market have been refunded due to event cancellation. [2017-08-02 11:30 PM][Di Wu Game 10 of Set 1-vs-Xin Gao Game 10 of Set 1][Match]",
    "createdDate": "2017-08-02 23:59:52"
  },
  {
    "text": "Wagers for the following market have been refunded due to event cancellation. [2017-08-02 11:30 PM][Di Wu Game 11 of Set 1-vs-Xin Gao Game 11 of Set 1][Match]",
    "createdDate": "2017-08-02 23:59:49"
  },
  {
    "text": "Due to a change in scores, wagers for the following market have been resettled, [2017-06-20 02:30 PM][Los Andes-vs-Gimnasia Jujuy][1st Half]",
    "createdDate": "2017-06-20 15:20:01"
  },
  {
    "text": "Wagers for the following market have been refunded due to event cancellation. [2017-06-15 11:00 AM][Bayern Munchen II (n)-vs-Wacker Innsbruck][Match]",
    "createdDate": "2017-06-15 10:39:28"
  },
  {
    "text": "The following wagers have been voided: Wager ID: 7540873, Cancellation Reason: nullDetails:  not verification \n",
    "createdDate": "2017-06-14 10:12:50"
  },
  {
    "text": "Due to a change in scores, wagers for the following market have been resettled, [2017-06-09 06:45 AM][Player to Win ATP French Open ? (All In)][Match]",
    "createdDate": "2017-06-11 20:43:32"
  }
]
```

3.16 FR007 – TRANSACTIONS 交易

This service will get Deposit/Withdrawal transactions made by Players under B2B Agent

这项服务将获得玩家在 B2B Agent 里做出的存取款交易

General 通用

Name	Value	Description
URL	/report/transactions	
Method	GET, POST	

Headers 头文件

Name	Value	Validation	Description
userCode	String (required 必需项)		This is agent code that you get at step 2. E.g: CO1AP1

			这是在 Step 2 里获取的 agent code,例如：CO1AP1
token	String (required 必需项)	*Token is available in **15 minutes** since it was created Token 令牌在创建之后的 15 分钟内有效*	

Parameters 参数

Name	Value	Validation	Description
dateFrom	Date (optional 非必需项)	Created Date Time format yyyy-MM-dd HH:mm:ss GMT-4 创建时间格式年-月-日-时-分-秒 GMT-4	*Example 例如: 2016-10-15 23:59:59*
dateTo	Date (optional 非必需项)	Created Date Time format yyyy-MM-dd HH:mm:ss GMT-4 创建时间格式年-月-日-时-分-秒 GMT-4	*Example 例如: 2016-10-16 23:59:59*
Note 注意: 1. **WITHOUT** date range 如没有设定时间范围: a. System shall response all transactions till now 系统将回应直至当前的所有交易			
transferType	String (optional 非必需项)	ALL DEPOSIT WITHDRAWAL 所有存取款 *(default 默认: **ALL**)*	Indicates the transfer type to get 注明转账类型来获取
userCode	String (optional 非必需项)		This is user code / loginID of player 这是玩家的用户名/登录名

Sample code (java) - *See HttpUtils class at Appendix*

示例代码（java）– 请参阅附录里的 HttpUtils class

```java
import java.io.IOException;
import java.util.HashMap;
import java.util.Map;

private static void getAnnouncements() throws IOException {
    Map params = new HashMap();
    params.put("dateFrom", "2017-04-24 02:10:12");
    Map headers = new HashMap();
    headers.put("userCode", " Q231");
    headers.put("token", "gvbLZb70DhMNXMLw3egvktPT3sfDWnDC5QI97MXHqK9FCFO1n6oepaXQivCIVpsDlth/jbR/qgUcpi2wHc62Tw==");

    String url = "http://paapiqat.oreo88.com/b2b/report/transactions";
    String result = HttpUtils.get(url, "", params, headers);
    System.out.println(result);
}
```

Sample code (C#) 示例代码（C#）

```csharp
using System;
using System.Net.Http;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace Transactions
{
    class Program
    {
        static void Main(string[] args)
        {
            Dictionary<string, string> parameters = new Dictionary<string, string>();
            parameters.Add("dateFrom", "2017-04-24 02:10:12");

            Dictionary<string, string> headers = new Dictionary<string, string>();
            headers.Add("userCode","Q231");
            headers.Add("token","DIwYQJZIYiosWrbXQe+TFdyMk6POZvbcM1KjxxQObZEn0+efzoMCb3i+PWr1ZFuj0UciR8w+qqo1M2hJ965Y9w==");

            Console.WriteLine(HttpUtils.GetRequest("http://paapiqat.oreo88.com/b2b/report/transactions", parameters, headers).Result);
            Console.ReadKey();
        }
    }
}
```

Response OK 返回 OK

```json
[
  {
    "agentCode": "Q231",
    "transferType": "DEPOSIT",
    "amount": 1,
    "userCode": "Q23100000A",
    "loginId": "Q23100000A",
    "remarks": "Deposit to Q23100000A",
    "transferDate": "2017-04-24 02:10:12"
  },
  {
    "agentCode": "Q231",
    "transferType": "WITHDRAWAL",
    "amount": 1,
    "userCode": "Q23100000A",
    "loginId": "Q23100000A",
    "remarks": "Withdrawal from Q23100000A",
    "transferDate": "2017-04-24 02:13:26"
  },
  {
    "agentCode": "Q231",
    "transferType": "DEPOSIT",
    "amount": 1000,
    "userCode": "Q231000000",
    "loginId": "Q231000000",
    "remarks": "Deposit to Q231000000",
    "transferDate": "2017-05-03 00:38:39"
  }

]
```

3.17 FR008 – Check Status For Deposit/Withdraw 查询存取款状态

This service will get status Deposit/Withdrawal transactions made by Players under B2B Agent

这项服务将获得玩家在 B2B Agent 里的存取款交易状态

General 通用

Name	Value	Description
URL	/player/depositwithdraw/status	
Method	GET, POST	

Headers 头文件

Name	Value	Validation	Description
userCode	String (required 必需项)		This is agent code that you get at step 2. E.g: CO1AP1 这是在 Step 2 里获取的 agent code,例如：CO1AP1
token	String (required 必需项)	*Token is available in **15 minutes** since it was created Token 令牌在创建之后的 15 分钟内有效*	

Parameters 参数

Name	Value	Validation	Description
transactionId	Long (required 必需项)	Value > 0	*transactionId you send at deposit/withdraw 您在调取存取款数据时发送的交易 ID Example 例如*

			12345678

Sample code (java) - *[See HttpUtils class at Appendix]*
示例代码（java） – 请参阅附录里的 HttpUtils class

```java
import java.io.IOException;
import java.util.HashMap;
import java.util.Map;

private static void getStatus() throws IOException {
    Map params = new HashMap();
    params.put("transactionId", "12345678");
    Map headers = new HashMap();
    headers.put("userCode", " Q231");
    headers.put("token", "gvbLZb70DhMNXMLw3egvktPT3sfDWnDC5QI97MXHqK9FCFO1n6oepaXQivCIVpsDlth/jbR/qgUcpi2wHc62Tw==");

    String url = "http://paapiqat.oreo88.com/b2b/player/depositwithdraw/status";
    String result = HttpUtils.get(url, "", params, headers);
    System.out.println(result);
}
```

Sample code (C#) 示例代码（C#）

```csharp
using System;
using System.Net.Http;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace ChangeStatusForDepositWithdraw
{
    class Program
    {
        static void Main(string[] args)
        {
            Dictionary<string, string> parameters = new Dictionary<string, string>();
            parameters.Add("transactionId", "12345678");

            Dictionary<string, string> headers = new Dictionary<string, string>();
            headers.Add("userCode","Q231");
            headers.Add("token","DIwYQJZIYiosWrbXQe+TFdyMk6POZvbcM1KjxxQObZEn0+efzoMCb3i+PWr1ZFuj0UciR8w+qqo1M2hJ965Y9w==");

            Console.WriteLine(HttpUtils.GetRequest("http://paapiqat.oreo88.com/b2b/player/depositwithdraw/status", parameters, headers).Result);
            Console.ReadKey();
        }
    }
}
```

Response OK 回应 OK

```json
{
    "status": "SUCCESS",
    "transferDate": "2018-11-15 23:18:34",
    "transferType": "DEPOSIT",
    "amount": 0.2,
    "userCode": "Q231000000"
}
```

```
{
    "status": "FAILED",
    "transferDate": "2018-11-14 20:15:20",
    "transferType": "WITHDRAW",
    "amount": 5,
    "userCode": "Q231000000"
}
```

```
{
    "status": "NOT_EXISTS"
}
```

3.18 FR009 – GET HOT EVENTS 获取热门赛事

This service will get hot events configured by B2B Agent.

这项服务将获得 B2B Agent 设置的热门赛事

General 通用

Name	Value	Description
URL	/v1/hot-events	
Method	GET, POST	

Headers 头文件

Name	Value	Validation	Description
userCode	String (required 必需项)		This is agent code that you get at step 2. E.g: CO1AP1 这是在 Step 2 里获取的 agent code,例如：CO1AP1

Parameters 参数

Name	Value	Validation	Description
locale 区域设置	String (optional 非必需项)	List supported locales based on available brand's languages. *在基于现有品牌的语言下，列出所支持区域*	Appendix Locale (Language) 参见附件区域设置（语言）
oddsFormat 赔率格式	String (optional 非必需项)	List supported oddsFormat: AM, EU, HK, ID, MY	Appendix Odds Format

		列出所支持赔率格式	参见附件赔率格式
sports	String (optional 非必需项)	The list of sport names, specified as the comma-separated	Appendix Sport

Sample code (java) - See HttpUtils class at Appendix
示例代码（java） – 请参阅附录里的 HttpUtils class

```java
import java.io.IOException;
import java.util.HashMap;
import java.util.Map;

private static void getHotEvents() throws IOException {
     Map params = new HashMap();
    params.put("sports", "soccer");
    params.put("locale", "en");
    params.put("oddsFormat", "EU");
    Map headers = new HashMap();
    headers.put("userCode", " Q231");

    String url = "http://paapiqat.oreo88.com/b2b/v1/hot-events";
    String result = HttpUtils.get(url, "", params, headers);
    System.out.println(result);
}
```

Sample code (C#) 示例代码（C#）

```csharp
using System;
using System.Net.Http;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace ChangeStatusForDepositWithdraw
{
    class Program
    {
        static void Main(string[] args)
        {
         Dictionary<string, string> parameters = new Dictionary<string, string>();
         parameters.Add("sports", "soccer");
         parameters.Add("locale", "en");
         parameters.Add("oddsFormat", "EU");
         Dictionary<string, string> headers = new Dictionary<string, string>();
         headers.Add("userCode","Q231");
         Console.WriteLine(HttpUtils.GetRequest("http://paapiqat.oreo88.com/b2b/v1/hot-events", parameters, headers).Result);
         Console.ReadKey();
        }
    }
}
```

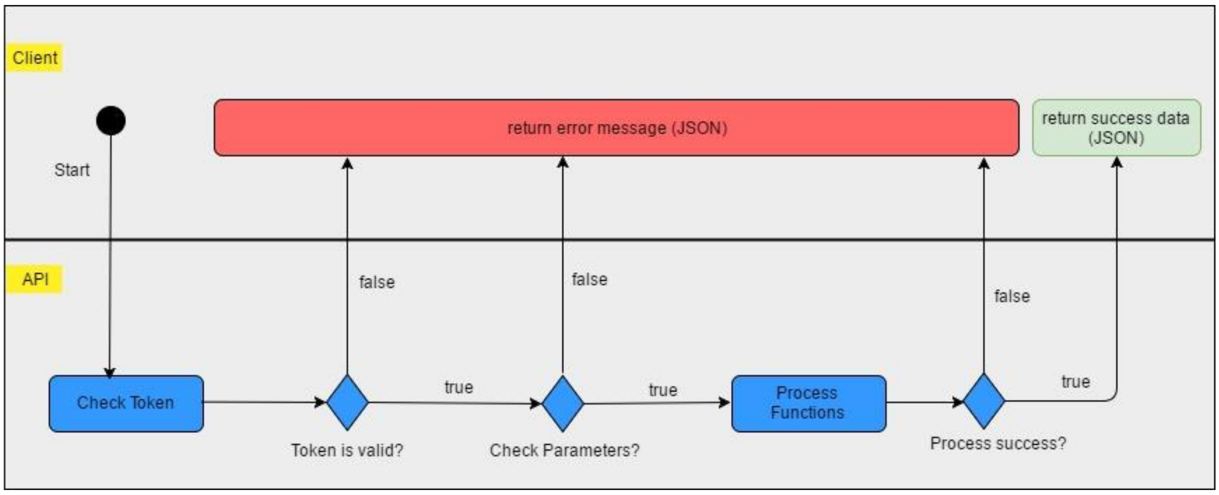
Response OK 回应 OK

```
  [{
        "id": 29,
        "name": "Soccer",
        "leagues": [{
                "id": 1728,
                "name": "Sweden - Allsvenskan",
                "events": [{
                        "id": 1000605833,
                        "home": "Djurgardens IF",
                        "away": "Malmo FF",
                        "starts": "2019-07-14T15:30:00Z",
                        "spreads": [{
                                "hdp": 0,
                                "home": -116,
                                "away": -102
                            }, {
                                "hdp": 0.5,
                                "home": -222,
                                "away": 177
                            }
                        ],
                        "moneyline": {
                            "home": 165,
                            "away": 181,
                            "draw": 231
                        },
                        "totals": [{
                                "points": 2.25,
                                "over": -114,
                                "under": -105
                            }, {
                                "points": 1.75,
                                "over": -230,
                                "under": 179
                            }
                        ]
                    }
                ]
            }
        ]
    }
]
```

4 SCREENS AND WORKFLOWS 截图和工作流程

Please see basic workflow in each API functions.

请参阅每个 API 函数的基本工作流程。



5 Appendix 附录

5.1 Errors Table 错误表格

Code	Message	Description
103	Your player's account has been <status>. 您用户的账号已经被 <状态>	
104	Member not exist in system. 系统没有这个用户	
105	User create fail. 创建用户失败	
106	User locked by login fail many times. 登录失败过多导致账户被锁	
107	Agent hasn't key store in system. 代理系统未储存密钥	
108	Agent invalid. Please contact partner to get agent code. 代理码无效。请联系合作伙伴获取代理码	
109	You don't allow create player under agent code who is not directly downline. 您无法创建不属于在该代理线下的玩家	
110	Agent not exist in system. 该代理账户不存在系统内	
111	Login id ready exist in system. 该登入名已被使用	
112	Login id is not valid. 登入名无效	
113	User's brand is not support login.	

	客户品牌不支持登陆	
114	Cannot change status when user is Closed or Suspended by Company. 若用户的账号被上线关闭或暂停使用，无法更改状态	
115	User account hasn't exist in system. 用户账号在系统里不存在	
116	You could not create more downline because you reached the maximum limitation. 您不能创建下线，因你已经达到最高上限	
305	Player has no permission create key store. 玩家不被许可创建密钥存储	
306	Invalid parameters. You will have information in message of error. 参数无效，您会收到错误信息。	
307	Account Balance do not exist in system. 账户余额不存在系统内。	
308	Your amount should be a positive number 您的金额应该是一个正数	
309	Your balance is not enough. 您的余额不足。	
310	Your balance exceeded credit limit. 您的余额超过了信用额度	
311	Amount value should be two decimal places. 金额值应该是小数点后两位。	
403	The token for this brand is still not generated. 这个品牌的代码还没有产生。	
405	Your wallet do not exist in system. 您的钱包在系统不存在。	
406	Your wallet is inactive. 您的钱包未激活	
407	Invalid product. 产品无效。	
423	Your account is <status>. Please contact your upline for help. 您的帐户是<状态>。请联系您的上线寻求帮助。	

5.2 WAGER STATUS 注单状态

Code	Name	Description
PENDING	Wager is pending 注单未结算	
OPEN	Wager is opening	

	等待中	
SETTLED	Wager was settled 注单已结算	
CANCELLED	Wager was cancelled 注单被取消	
DELETED	Wager was deleted 注单被删除	

5.3 Bet Type 投注类型

Code	Name	Description
1	ML_1X2	1x2 独赢盘
2	HDP	Handicap 让球盘
3	OU	Over Under 大小盘
4	HOME_TOTALS	主队_总得分
5	AWAY_TOTALS	客队_总得分
6	MIX_PARLAY	混合过关
7	TEASER	变让分过关盘
8	MANUAL_PLAY	人工注單
97	OE	Odd Even 单双
99	OUTRIGHT	优生冠军

5.4 Selection Type 选择类型

Code	Name	Description
0	HOME 主队	Home team is selected 选定"主队"
1	AWAY 客队	Away team is selected 选定"客队"
2	DRAW 和局	Draw is selected 选定"和局"
3	OVER 大	Over is selected 选定"大"
4	UNDER 小	Under is selected 选定"小"
5	HOME_OVER 主队大	Home over is selected 选定"主队大"
6	HOME_UNDER 主队小	Home under is selected 选定"主队小"
7	AWAY_OVER 客队大	Away over is selected 选定"客队大"
8	AWAY_UNDER 客队小	Away under is selected 选定"客队小"
9	ODD 单数	Odd type is selected 选定"单数"
10	EVEN 双数	Even type is selected 选定"双数"
11	TEAM 队伍	Team type is selected 选定"队伍"

5.5 ODDS FORMAT 赔率格式

Code 代码	Name 名称	Description
0	AM 美式盘	American odds format 美式盘格式
1	EU 欧洲盘	Euro odds format 欧洲盘格式
2	HK 香港盘	Hong Kong odds format 香港盘格式
3	ID 印尼盘	Indo odds format 印尼盘格式
4	MY 马来盘	Malay odds format 马来盘格式

5.6 USER STATUS 用户状态

Code 代码	Name 名称	Description 描述
ACTIVE	User is active 用户账号已激活	**'ACTIVE'** player **CAN** login and place bet on Member site **'活跃'**用户**可以**在客户网站登录和进行投注
INACTIVE	User is inactive 用户账号未激活	**'INACTIVE'** player **CANNOT** login on Member site **'不活跃'**用户**不可以**在客户网站登录
SUSPENDED	User is suspended by direct agent. 用户被上线代理暂停	**'SUSPENDED'** player **CAN** login on Member site, but **CANNOT** place bet **'暂停'**用户**可以**在客户网站登录但是不可以投注
SUSPENDED_BY_COMPANY	User is suspended by 'Company' agent. 用户被公司级代理暂停使用	**'SUSPENDED_BY_COMPANY'** player **CAN** login on Member site, but **CANNOT** place bet **'公司代理暂停'**用户**可以**在客户网站登录但是不可以投注
CLOSED	User is closed by administrator 用户账号被管理员关闭	**'CLOSED'** player **CANNOT** login on Member site. Agent **CANNOT** update any information for Closed player.

		'关闭'用户**不可以**在客户网站登录但是不可以投注。代理不可以更新任何关闭用户的信息。

5.7 Period Status 期间状态

Code 代码	Name 名称	Description 描述
0	Badminton 羽毛球	Match 赛事
1	Badminton 羽毛球	1st Game_Badminton 第一场比赛_羽毛球
2	Badminton 羽毛球	2nd Game_Badminton 第二场比赛_羽毛球
3	Badminton 羽毛球	3rd Game_Badminton 第三场比赛_羽毛球
0	Bandy 班迪球	Match 赛事
1	Bandy 班迪球	1st Half 上半场
2	Bandy 班迪球	2nd Half 下半场
0	Baseball 棒球	Game 赛事
1	Baseball 棒球	1st Half 上半场
2	Baseball 棒球	2nd Half 下半场
3	Baseball 棒球	1st Inning 第一局
4	Baseball 棒球	2nd Inning 第二局
0	Basketball 篮球	Game 赛事
1	Basketball 篮球	1st Half 上半场
2	Basketball 篮球	2nd Half 下半场
3	Basketball 篮球	1st Quarter 第一节
4	Basketball 篮球	2nd Quarter 第二节
5	Basketball 篮球	3rd Quarter 第三节
6	Basketball 篮球	4th Quarter 第四节
0	Beach Volleyball 沙滩排球	Match 赛事
1	Beach Volleyball 沙滩排球	1st Set 首盘
2	Beach Volleyball 沙滩排球	2nd Set 第二盘
3	Beach Volleyball 沙滩排球	3rd Set 第三盘
0	Boxing 拳击	Fight 拳击
0	Chess 西洋棋	Match 赛事
0	Cricket 板球、木球	Match 赛事
1	Cricket 板球、木球	1st Inning 第一局
2	Cricket 板球、木球	2nd Inning 第二局
0	Curling 冰壶	Game 赛事
1	Curling 冰壶	1st End 第一场结束

0	Darts 飞镖	Match 赛事
1	Darts 飞镖	1st Set 首盘
2	Darts 飞镖	2nd Set 第二盘
3	Darts 飞镖	3rd Set 第三盘
4	Darts 飞镖	4th Set 第四盘
5	Darts 飞镖	5th Set 第五盘
0	Darts (Legs) 飞镖 （局)	Match 赛事
1	Darts (Legs) 飞镖 （局)	1st Leg 第一局
2	Darts (Legs) 飞镖 （局)	2nd Leg 第二局
3	Darts (Legs) 飞镖 （局)	3rd Leg 第三局
4	Darts (Legs) 飞镖 （局)	4th Leg 第四局
5	Darts (Legs) 飞镖 （局)	5th Leg 第五局
6	Darts (Legs) 飞镖 （局)	6th Leg 第六局
0	E Sports 电子竞技	Match 赛事
1	E Sports 电子竞技	Map 1 地图一
2	E Sports 电子竞技	Map 2 地图二
3	E Sports 电子竞技	Map 3 地图三
4	E Sports 电子竞技	Map 4 地图四
5	E Sports 电子竞技	Map 5 地图五
6	E Sports 电子竞技	Map 6 地图六
7	E Sports 电子竞技	Map 7 地图七
0	Field Hockey 曲棍球	Match 赛事
1	Field Hockey 曲棍球	1st Half 上半场
2	Field Hockey 曲棍球	2nd Half 下半场
0	Floorball 福樂球、地板球	Match 赛事
1	Floorball 福樂球、地板球	1st Period 第一节
2	Floorball 福樂球、地板球	2nd Period 第二节
3	Floorball 福樂球、地板球	3rd Period 第三节
0	Football 橄榄球	Game 赛事
1	Football 橄榄球	1st Half 上半场
2	Football 橄榄球	2nd Half 下半场
3	Football 橄榄球	1st Quarter 第一节
4	Football 橄榄球	2nd Quarter 第二节
5	Football 橄榄球	3rd Quarter 第三节
6	Football 橄榄球	4th Quarter 第四节
0	Futsal 室内足球	Match 赛事
1	Futsal 室内足球	1st Half 上半场
2	Futsal 室内足球	2nd Half 下半场
0	Golf 高尔夫球	Matchups 对决
0	Handball 手球	Match 赛事

1	Handball 手球	1st Half 上半场
2	Handball 手球	2nd Half 下半场
0	Hockey 冰上曲棍球	Game 赛事
1	Hockey 冰上曲棍球	1st Period 第一节
2	Hockey 冰上曲棍球	2nd Period 第二节
3	Hockey 冰上曲棍球	3rd Period 第三节
0	Horse Racing 赛马	Race 赛事
0	Lacrosse 袋棍球	Match 全场
1	Lacrosse 袋棍球	1st Half 上半场
2	Lacrosse 袋棍球	2st Half 下半场
3	Lacrosse 袋棍球	1st Quarter 第一节
4	Lacrosse 袋棍球	2nd Quarter 第二节
5	Lacrosse 袋棍球	3rd Quarter 第三节
6	Lacrosse 袋棍球	4th Quarter 第四节
0	Mixed Martial Arts 综合格斗	Fight 赛事
1	Mixed Martial Arts 综合格斗	Round 1 第一局
2	Mixed Martial Arts 综合格斗	Round 2 第二局
3	Mixed Martial Arts 综合格斗	Round 3 第三局
4	Mixed Martial Arts 综合格斗	Round 4 第四局
5	Mixed Martial Arts 综合格斗	Round 5 第五局
0	Other Sports 其它	Game 赛事
0	Politics 政治	Election 选举
0	Rink Hockey 滚轴曲棍球	Match 赛事
1	Rink Hockey 滚轴曲棍球	1st Period 第一节
2	Rink Hockey 滚轴曲棍球	2nd Period 第二节
0	Rugby League 联盟式橄榄球	Match 赛事
1	Rugby League 联盟式橄榄球	1st Half 上半场
2	Rugby League 联盟式橄榄球	2nd Half 下半场
0	Rugby Union 联合式橄榄球	Match 赛事
1	Rugby Union 联合式橄榄球	1st Half 上半场
2	Rugby Union 联合式橄榄球	2nd Half 下半场
0	Snooker 桌球、撞球	Match 赛事
1	Snooker 桌球、撞球	1st Frame 第一回合
0	Soccer 足球	Match 赛事
1	Soccer 足球	1st Half 上半场
2	Soccer 足球	2nd Half 下半场
0	Softball 垒球	Game 赛事
1	Softball 垒球	1st Half 上半场
2	Softball 垒球	2st Half 下半场
0	Squash 壁球	Match 赛事

1	Squash 壁球	1st Game 第一场比赛
2	Squash 壁球	2nd Game 第二场比赛
3	Squash 壁球	3rd Game 第三场比赛
4	Squash 壁球	4th Game 第四场比赛
5	Squash 壁球	5th Game 第五场比赛
0	Table Tennis 乒乓球	Match 赛事
1	Table Tennis 乒乓球	1st Game 第一场比赛
2	Table Tennis 乒乓球	2nd Game 第二场比赛
3	Table Tennis 乒乓球	3rd Game 第三场比赛
4	Table Tennis 乒乓球	4th Game 第四场比赛
5	Table Tennis 乒乓球	5th Game 第五场比赛
6	Table Tennis 乒乓球	6th Game 第六场比赛
0	Tennis 网球	Match 赛事
1	Tennis 网球	1st Set Winner 首盘优胜者
2	Tennis 网球	2nd Set Winner 第二盘优胜者
3	Tennis 网球	3rd Set Winner 第三盘优胜者
4	Tennis 网球	4th Set Winner 第四盘优胜者
5	Tennis 网球	5th Set Winner 第五盘优胜者
0	Volleyball 排球	Match 赛事
1	Volleyball 排球	1st Set 首盘
2	Volleyball 排球	2nd Set 第二盘
3	Volleyball 排球	3rd Set 第三盘
4	Volleyball 排球	4th Set 第四盘
5	Volleyball 排球	5th Set 第五盘
0	Volleyball (Points) 排球（得分)	Game 赛事
1	Volleyball (Points) 排球（得分)	1st Set 首盘
2	Volleyball (Points) 排球（得分)	2nd Set 第二盘
3	Volleyball (Points) 排球（得分)	3rd Set 第三盘
4	Volleyball (Points) 排球（得分)	4th Set 第四盘
5	Volleyball (Points) 排球（得分)	5th Set 第五盘
0	Water Polo 水球	Match 赛事
1	Water Polo 水球	1st Period 第一节
2	Water Polo 水球	2nd Period 第二节
3	Water Polo 水球	3rd Period 第三节
4	Water Polo 水球	4th Period 第四节
0	Padel Tennis 板网球	Match 赛事
1	Padel Tennis 板网球	1st Set Winner 首盘优胜者
2	Padel Tennis 板网球	2nd Set Winner 第二盘
3	Padel Tennis 板网球	3rd Set Winner 第三盘
0	Aussie Rules Footbal 澳洲足球	Game 赛事

1	Aussie Rules Footbal 澳洲足球	1st Half 上半场
2	Aussie Rules Footbal 澳洲足球	2nd Half 下半场
3	Aussie Rules Footbal 澳洲足球	1st Quarter 第一节
4	Aussie Rules Footbal 澳洲足球	2nd Quarter 第二节
5	Aussie Rules Footbal 澳洲足球	3rd Quarter 第三节
6	Aussie Rules Footbal 澳洲足球	4th Quarter 第四节
0	Aussie Rules 澳洲足球	Game 赛事
1	Aussie Rules 澳洲足球	1st Half 上半场
2	Aussie Rules 澳洲足球	2nd Half 下半场
3	Aussie Rules 澳洲足球	1st Quarter 第一节
4	Aussie Rules 澳洲足球	2nd Quarter 第二节
5	Aussie Rules 澳洲足球	3rd Quarter 第三节
6	Aussie Rules 澳洲足球	4th Quarter 第四节
0	Alpine Skiing 高山滑雪	Matchups 对决
0	Biathlon 冬季两项	Matchups 对决
0	Ski Jumping 跳台滑雪	Matchups 对决
0	Cross Country 越野滑雪	Matchups 对决
0	Formula 1 一级方程式赛车	Matchups 对决
0	Cycling 自行车	Matchups 对决
0	Bobsleigh 有舵雪橇	Matchups 对决
0	Figure Skating 花样滑冰	Matchups 对决
0	Freestyle Skiing 自由式滑雪	Matchups 对决
0	Luge 无舵雪橇	Matchups 对决
0	Nordic Combined 北欧两项	Matchups 对决
0	Short Track 短道速滑	Matchups 对决
0	Skeleton 俯式冰橇	Matchups 对决
0	Snow Boarding 单板滑雪	Matchups 对决
0	Speed Skating 速度滑冰	Matchups 对决

5.8 Sport 体育

Name 名称	Description 描述
soccer	Soccer 足球
tennis	Tennis 网球
basketball	Basketball 篮球
football	Football 美式足球
baseball	Baseball 棒球
golf	Golf 高尔夫球
hockey	Hockey 曲棍球
volleyball	Volleyball 排球

rugby-league	Rugby league 橄榄球联赛
mixed-martial-arts	Mixed martial arts 混合武术
handball	Handball 手球
e-sports	E-sports 电子竞技
aussie-rules	Aussie rules 澳洲规则
boxing	Boxing 拳击
rugby-union	Rugby union 橄榄球联盟
snooker	Snooker 斯诺克
cycling	Cycling 自行车赛
alpine-skiing	Alpine Skiing 高山滑雪
archery	Archery 射箭
athletics	Athletics 田径
badminton	Badminton 羽毛球
bandy	Bandy 往復投擲
beach-volleyball	Beach Volleyball 沙灘排球
biathlon	Biathlon 冬季兩項
bobsleigh	Bobsleigh 有舵雪橇
cricket	Cricket 板球
cross-country	Cross Country 越野
crossfit	Crossfit 混合體能
curling	Curling 冰壺
darts	Darts 飛鏢
darts-legs	Darts (Legs) 飛鏢（腿）
drone-racing	Drone Racing 無人機競賽
entertainment	Entertainment 綜藝歌唱節目
field-hockey	Field Hockey 曲棍球
figure-skating	Figure Skating 花樣溜冰
floorball	Floorball 福樂球
formula-1	Formula 1 一級方程式
freestyle-skiing	Freestyle Skiing 自由式滑雪
futsal	Futsal 五人制足球
horse-racing	Horse Racing 跑馬
luge	Luge 無舵雪橇
nordic-combined	Nordic Combined 北歐兩項
olympics	Olympics 奧林匹克
poker	Poker 撲克
politics	Politics 政治
short-track	Short Track 短道
skeleton	Skeleton 骨架
ski-jumping	Ski Jumping 跳台滑雪

snow-boarding	Snow Boarding 滑雪板
softball	Softball 壘球
speed-skating	Speed Skating 速滑
squash	Squash 壁球
table-tennis	Table Tennis 乒乓球
volleyball-points	Volleyball (Points) 排球（點）
water-polo	Water Polo 水球

5.9 LANGUAGE 语言

Locale code 区域代码	Description 描述
en	English
zh-cn	Simplified Chinese (简体中文)
zh-tw	Traditional Chinese (繁體中文)
id	Indonesian
vi	Vietnamese (Tiếng Việt)
ja	Japanese (日本語)
ko	Korean (한국어)
th	Thai (ภาษา ไทย)
km	Khmer (ភាសាខ្មែរ)
fr	French (Français)
de	German (Deutsch)
es	Spanish (Español)
he	Hebrew (עברית)
pt	Brazilian Portuguese
ru	Russian (Русский)

5.10 SAMPLE CODE HTTPUTILS (JAVA) 示例代码 HTTPUTILS (JAVA)

示例代码 **HTTPUTILS** *（JAVA）*

```
package com.demo.restful;

import org.apache.http.HttpEntity;
import org.apache.http.NameValuePair;
import org.apache.http.client.config.RequestConfig;
import org.apache.http.client.methods.CloseableHttpResponse;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.client.methods.HttpUriRequest;
import org.apache.http.config.Registry;
import org.apache.http.config.RegistryBuilder;
import org.apache.http.conn.socket.ConnectionSocketFactory;
import org.apache.http.conn.socket.LayeredConnectionSocketFactory;
import org.apache.http.conn.socket.PlainConnectionSocketFactory;
import org.apache.http.conn.ssl.SSLConnectionSocketFactory;
import org.apache.http.conn.ssl.SSLInitializationException;
```

```java
import org.apache.http.entity.StringEntity;
import org.apache.http.impl.client.CloseableHttpClient;
import org.apache.http.impl.client.HttpClients;
import org.apache.http.impl.conn.PoolingHttpClientConnectionManager;
import org.apache.http.message.BasicNameValuePair;
import org.apache.http.util.EntityUtils;

import java.io.IOException;
import java.net.URLEncoder;
import java.security.KeyManagementException;
import java.security.NoSuchAlgorithmException;
import java.util.ArrayList;
import java.util.List;
import java.util.Map;
import javax.net.ssl.SSLContext;
import org.apache.commons.lang.StringUtils;
import org.apache.http.client.config.CookieSpecs;

public final class HttpUtils {

    final static PoolingHttpClientConnectionManager CONNMGR;
    final static CloseableHttpClient CLIENT;
    private static final int MAX_PER_ROUTE = 200;
    private static final int MAXTOTAL = 200;
    private static final int VALIDATE_AFTER_INACTIVITY = 1000;
    private static final int CONNECTION_TIMEOUT = 60000;

    static {
        LayeredConnectionSocketFactory ssl = null;
        try {
            ssl = SSLConnectionSocketFactory.getSystemSocketFactory();
        } catch (final SSLInitializationException ex) {
            final SSLContext sslcontext;
            try {
                sslcontext = SSLContext.getInstance(SSLConnectionSocketFactory.TLS);
                sslcontext.init(null, null, null);
                ssl = new SSLConnectionSocketFactory(sslcontext);
            } catch (final SecurityException ignore) {
            } catch (final KeyManagementException ignore) {
            } catch (final NoSuchAlgorithmException ignore) {
            }
        }

        RequestConfig config = RequestConfig.custom()
                .setConnectTimeout(CONNECTION_TIMEOUT)
                .setConnectionRequestTimeout(CONNECTION_TIMEOUT)
                .setCookieSpec(CookieSpecs.STANDARD)
                .setSocketTimeout(CONNECTION_TIMEOUT).build();

        final Registry<ConnectionSocketFactory> sfr = RegistryBuilder.<ConnectionSocketFactory>
create()
                .register("http", PlainConnectionSocketFactory.getSocketFactory())
                .register("https", ssl != null ? ssl : SSLConnectionSocketFactory.getSocketFact
ory())
                .build();

        CONNMGR = new PoolingHttpClientConnectionManager(sfr);
        CONNMGR.setDefaultMaxPerRoute(MAX_PER_ROUTE);
        CONNMGR.setMaxTotal(MAXTOTAL);
        CONNMGR.setValidateAfterInactivity(VALIDATE_AFTER_INACTIVITY);
        CLIENT = HttpClients.custom().setConnectionManager(CONNMGR).setDefaultRequestConfig(con
fig).build();
    }

    public HttpUtils() {
    }

    public static String get(String url, String authorization, Map<String, String> queryParams) throws IOException {
        List<NameValuePair> params = new ArrayList<>();
        for (Map.Entry<String, String> entry : queryParams.entrySet()) {
            params.add(new BasicNameValuePair(entry.getKey(), entry.getValue()));
        }
        return get(url, authorization, params, null);
    }

    public static String get(String url, String authorization, Map<String, String> queryParams,
            Map<String, String> headers) throws IOException {
        List<NameValuePair> params = new ArrayList<>();
        for (Map.Entry<String, String> entry : queryParams.entrySet()) {
            params.add(new BasicNameValuePair(entry.getKey(), entry.getValue()));
        }
        return get(url, authorization, params, headers);
    }

    public static String get(String url, String authorization, List<NameValuePair> queryParams) throws IOException {
        return get(url, authorization, queryParams, null);
    }

    public static String get(String url, String authorization, List<NameValuePair> queryParams,
            Map<String, String> headers) throws IOException {
        CloseableHttpResponse response = null;
        StringBuilder requestParams = new StringBuilder();
        for (NameValuePair queryParam : queryParams) {
            requestParams.append(URLEncoder.encode(queryParam.getName(), "UTF-8")).append("=").append(URLEncoder.encode(String.valueOf(queryParam.getValue()), "UTF-8")).append("&");
        }

        if (requestParams.capacity() > 0) {
            url = url + "?" + StringUtils.stripEnd(requestParams.toString(), "&");
        }
        HttpUriRequest request = new HttpGet(url);
        // add header for request.
        if (headers != null && !headers.isEmpty()) {
            for (Map.Entry<String, String> i : headers.entrySet()) {
                request.addHeader(i.getKey(), i.getValue());
            }
        }
        request.addHeader("Authorization", authorization);
        request.addHeader("Accept", " application/json");
        request.addHeader("Accept-Encoding", "gzip, deflate");
        response = CLIENT.execute(request);
        HttpEntity entity = response.getEntity();

        String responseString = "";
        if (entity != null) {
            responseString = EntityUtils.toString(entity, "UTF-8");
        }

        return responseString;
    }

    public static String post(String url, String authorization, String requestPayload) throws IOException {
        HttpPost httpPost = new HttpPost(url);
        httpPost.addHeader("Authorization", authorization);
        httpPost.addHeader("Accept", " application/json");
        httpPost.setHeader("Content-type", "application/json");
        StringEntity strEntity = new StringEntity(requestPayload);
        httpPost.setEntity(strEntity);
        CloseableHttpResponse response = CLIENT.execute(httpPost);
        HttpEntity entity = response.getEntity();
        String responseString = EntityUtils.toString(entity, "UTF-8");
        return responseString;
    }
    public static String post(String url, String authorization, Map<String, String> queryParams
,
            Map<String, String> headers) throws IOException {
        StringBuilder requestParams = new StringBuilder();
        if (queryParams != null && !queryParams.isEmpty()) {
            for (Map.Entry<String, String> queryParam : queryParams.entrySet()) {
                requestParams.append(URLEncoder.encode(queryParam.getKey(), "UTF-
8")).append("=").append(URLEncoder.encode(String.valueOf(queryParam.getValue()), "UTF-
8")).append("&");
            }
        }
        if (requestParams.capacity() > 0) {
            url = url + "?" + StringUtils.stripEnd(requestParams.toString(), "&");
        }
        HttpPost httpPost = new HttpPost(url);

        httpPost.addHeader("Authorization", authorization);
        httpPost.addHeader("Accept", " application/json");
        httpPost.setHeader("Content-type", "application/json");
        // add header for request.
        if (headers != null && !headers.isEmpty()) {
            for (Map.Entry<String, String> i : headers.entrySet()) {
                httpPost.addHeader(i.getKey(), i.getValue());
            }
        }
        CloseableHttpResponse response = CLIENT.execute(httpPost);
        HttpEntity entity = response.getEntity();
        String responseString = EntityUtils.toString(entity, "UTF-8");
        return responseString;
    }
}
```